LREC-COLING 2024

Third Workshop on Language Technologies for Historical and Ancient Languages @LREC-COLING-2024 (LT4HALA 2024)

Workshop Proceedings

Editors
Rachele Sprugnoli and Marco Passarotti

25 May, 2024
Torino, Italia

Proceedings of LT4HALA 2024: The Third Workshop on Language Technologies for Historical and Ancient Languages @LREC-COLING-2024

ISBN 978-2-493814-46-3
ISSN 2951-2093 (COLING); 2522-2686 (LREC)

Jointly organized by the ELRA Language Resources Association
and the International Committee on Computational Linguistics

Preface

These proceedings include the papers accepted for presentation at the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2024).[1] The workshop was held on May 25th 2024 in Turin, Italy, co-located with the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024).[2]

The workshop wants to provide a venue to discuss research works on a wide range of topics concerning the building, analysis, exploitation and distribution of collections of digitized texts written in historical and ancient languages as well as of their lexica, with a specific focus on the development and application of Language Technologies (LTs) for such purposes.

The topics of the workshop are strictly bound to the peculiar characteristics of textual and lexical data for historical and ancient languages, which set them apart from modern languages, with a significant impact on the use and development of LTs for their processing and study. Among the topics covered by the workshop are issues about the digitization process of textual sources, like handling spelling variation, and detecting and correcting OCR (Optical Character Recognition) errors. Issues related to the automatic processing of various layers of metalinguistic annotation are also included. Annotation is made complex by the sparsity and inconsistency of texts that present considerable orthographic variation, are sometimes incomplete and belong to a large spectrum of literary genres. Such issues raise problems of adaptation of Natural Language Processing (NLP) tools and pipelines to address diachronic/diatopic/diastratic variation in texts, which requires to be properly evaluated.

The various LTs tasks related to the topics of LT4HALA require a strict collaboration between scholars from different disciplinary areas. In such respect, the objective of the LT4HALA workshop series is to foster cross-fertilization between the Computational Linguistics community and the areas in the Humanities dealing with historical linguistic data, e.g. historians, philologists, linguists, archaeologists and literary scholars. Such a wide and diverse range of disciplines and scholars involved in the development and use of LTs for historical and ancient languages is mirrored by the large set of topics covered by the papers published in these proceedings, which, among others, include the creation and evaluation of annotated corpora and lexical resources for historical languages, and the use of Large Language Models (together with their fine-tuning) for performing various NLP tasks, like machine translation, summarization, sentiment analysis, dependency parsing, part-of-speech tagging, named entity recognition, and authorship attribution.

As large as the number of topics discussed in the papers is that of the either ancient/dead languages or the historical varieties of modern/living ones concerned. Overall, the languages tackled in the papers published in these proceedings are the following: Latin (as the most represented language), Old English, Old Irish, Old Italian, Dutch (in historical documents), Middle French, Ancient Greek, Hebrew, XIX century Italian and English, variations of the Ancient Egyptian languages (Old, Middle, and Late Egyptian, Demotic, Coptic), Gothic, Classical Armenian, Old High German.

In the call for papers, we invited to submit proposals of different types, such as experimental papers, reproduction papers, resource papers, position papers and survey papers. We asked both for long and short papers describing original and unpublished work. We defined as suitable long papers (up to 9 pages, plus references) those that describe substantial completed research and/or report on the development of new methodologies, as well as position papers. Short papers (up to 5 pages, plus references) were instead more appropriate for reporting on works in progress or for describing a specific tool or project. We encouraged the authors

[1] https://circse.github.io/LT4HALA/2024/

[2] https://lrec-coling-2024.org

of papers reporting experimental results to make their results reproducible and the entire process of analysis replicable, by distributing the data and the tools they used. Like for LREC-COLING 2024, the submission process was double-blind. Each paper was reviewed but three independent reviewers from a program committee made of 27 scholars (13 women and 14 men) from 13 countries. In total, we received 32 submissions (against the 24 of the previous edition). After the reviewing process, we accepted 20 submissions, leading to an acceptance rate of 62.50%.

LT4HALA 2024 was also the venue of the third edition of EvaLatin, the campaign devoted to the evaluation of NLP tools for Latin.[3] EvaLatin was started in 2020 (co-located with the first edition of LT4HALA) considering the important role played by textual data and linguistic metadata in the study of historical and ancient languages, with a special focus on Latin due to its prominence among such languages, both for the size and for the degree of diversity of its texts. Running evaluation campaigns in such a scenario is essential to understand the level of accuracy of the NLP tools used to build and analyze resources featuring texts that show those peculiar characteristics mentioned above. The third edition of EvaLatin focused on two shared tasks (i.e. Dependency Parsing, and Emotion Polarity Detection). The Dependency Parsing task was based on the Universal Dependencies (UD) framework.[4] No specific training data was released but participants were left free to make use of any (kind of) resource they consider useful for the task, including the Latin treebanks already available in the UD collection. In this regard, one of the challenges of this task was to understand which treebank (or combination of treebanks) is the most suitable to deal with new test data. Test data included both prose and poetic texts. Also for the Emotion Polarity Detection task, no training data were released but participants were provided with an annotation sample, a manually created polarity lexicon and annotation guidelines. Again, participants were left free to pursue the approach they prefer, including unsupervised and/or cross-language ones. Test data were poetic texts from different time periods. Shared data and all the necessary evaluation scripts were distributed to participants. Participants were required to submit a technical report for each task (with all the related sub-tasks) they took part in. The maximum length of the reports was 4 pages (plus references). In total, these proceedings include 5 technical reports of EvaLatin, corresponding to as many participants (3 for the Dependency Parsing Task, and 2 for the Emotion Polarity Detection task). All reports received a light review by the organizers who checked the correctness of the format, the exactness of the results and ranking reported, as well as the overall exposition. The proceedings also feature a paper detailing some specific aspects of the third edition of EvaLatin, like dataset, annotation criteria and results of the shared tasks.

Besides EvaLatin, LT4HALA 2024 hosted also the third edition of EvaHan, an evaluation campaign of NLP tools for the Ancient Chinese language, organized by a team of scholars directed by Bin Li (Nanjing Normal University)[5] The third edition of EvaHan focused on one task, namely a joint task of Sentence Segmentation and Punctuation. The EvaHan 2024 dataset was made of texts from classical sources, notably Siku Quanshu, along with other historical texts. The dataset's processing involved initial automatic punctuation and sentence segmentation. Subsequently, these automatic outputs were corrected and refined by experts in Ancient Chinese language to ensure the highest quality of gold standard texts. All evaluation data were txt files in Unicode (UTF-8) format. The training data comprised 10 million characters sourced from the Siku Quanshu. The test data included approximately 50,000 characters of Ancient Chinese texts. Participants were allowed to submit runs following two modalities. In the closed modality, each team was allowed to use only the training data provided, and the pre-trained model XunziALLM, which is a large language model for ancient Chinese processing. In the open modality, there was no limit on the resources, data and models: annotated external

[3] https://circse.github.io/LT4HALA/2024/EvaLatin

[4] https://universaldependencies.org

[5] https://circse.github.io/LT4HALA/2024/EvaHan

data, such as the components or Pinyin of the Chinese characters, or word embeddings could be employed. Like for EvaLatin, the participants of EvaHan were required to submit a short technical report which received a light review by the organizers. Overall, these proceedings include an overview of the EvaHan campaign (authored by the organizers) and 6 technical reports, corresponding to as many participants.

We are grateful to the organizers of EvaHan, who contributed to extend the range of historical and ancient languages of the LT4HALA 2024 workshop and showed how some NLP-related issues concern ancient and historical languages per se, despite their typological differences.

Now in its third edition, LT4HALA is constantly growing both as for the number of participants and as for the quantity and diversity of the languages and topics addressed by their scholarly contributions. We are glad to realize that the field is getting bigger, yet considering that this is not surprising, as the study of ancient and historical languages has always been strictly bound to the analysis of the empirical evidence provided by texts. Processing the collections of such texts, which today are largely available in digital format, by using the most advanced LTs to perform their computational analysis, promises to advance the state of the art in the century-long study of our linguistic past. LT4HALA wants to provide a venue to support such a computational turn.

Rachele Sprugnoli
Marco Passarotti

Workshop Organizers:

Rachele Sprugnoli, Università degli Studi di Parma (Italy)
Marco Passarotti, Università Cattolica del Sacro Cuore di Milano (Italy)

Program Committee:

Adam Anderson, FactGrid Cuneiform Project (USA)
Yannis Assael, Google DeepMind (UK)
Monica Berti, University of Leipzig (Germany)
Luca Brigada Villa, Università di Bergamo (Italy)
Flavio Massimiliano Cecchini, Katholieke Universiteit Leuven (Belgium)
Claudia Corbetta, Università degli Studi di Bergamo (Italy)
Margherita Fantoli, Katholieke Universiteit Leuven (Belgium)
Federica Gamba, Charles University (Czech Republic)
Shai Gordin, Ariel University (Israel)
Timo Korkiakangas, University of Helsinki (Finland)
Federica Iurescia, Università Cattolica del Sacro Cuore di Milano (Italy)
Bin Li, Nanjing Normal University (P.R. China)
Eleonora Litta, Università Cattolica del Sacro Cuore di Milano (Italy)
Yudong Liu, Western Washington University (USA)
Francesco Mambrini, Università Cattolica del Sacro Cuore di Milano (Italy)
Barbara McGillivray, Turing Institute (UK)
Chiara Palladino, Furman University (USA)
John Pavlopoulos, Athens University of Economics and Business (Greece)
Giulia Pedonese, Istituto di Linguistica Computazionale, CNR-ILC (Italy)
Matteo Pellegrini, Università Cattolica del Sacro Cuore di Milano (Italy)
Eva Pettersson, Uppsala University (Sweden)
Sophie Prévost, Laboratoire Lattice (France)
Thea Sommerschield, University of Nottingham (UK)
James Tauber, Eldarion (USA)
Alan Thomas, University of Sheffield (UK)
Toon Van Hal, Katholieke Universiteit Leuven (Belgium)
Tariq Yousef, University of Southern Denmark (Denmark)

EvaLatin 2024 Organizers:

Rachele Sprugnoli, Università degli Studi di Parma (Italy)
Federica Iurescia, Università Cattolica del Sacro Cuore di Milano (Italy)
Marco Passarotti, Università Cattolica del Sacro Cuore di Milano (Italy)

EvaHan 2024 Organizers:

Bin Li, Nanjing Normal University (P.R. China)
Bolin Chang, Nanjing Normal University (P.R. China)
Minxuan Feng, Nanjing Normal University (P.R. China)
Chao Xu, Nanjing Normal University (P.R. China)
Dongbo Wang, Nanjing Agricultural University (P.R. China)

Table of Contents

Workshop Program

Saturday, May 25, 2024

+ ***Long and short workshop papers***

Goidelex: A Lexical Resource for Old Irish
Cormac Anderson, Sacha Beniamine and Theodorus Fransen

Developing a Part-of-speech Tagger for Diplomatically Edited Old Irish Text
Adrian Doyle and John P. McCrae

From YCOE to UD: Rule-based Root Identification in Old English
Luca Brigada Villa and Martina Giarda

Too Young to NER: Improving Entity Recognition on Dutch Historical Documents
Vera Provatorova, Marieke van Erp and Evangelos Kanoulas

Towards Named-Entity and Coreference Annotation of the Hebrew Bible
Daniel G. Swanson, Bryce D. Bussert and Francis Tyers

LiMe: A Latin Corpus of Late Medieval Criminal Sentences
Alessanda Clara Carmela Bassani, Beatrice Giovanna Maria Del Bo, Alfio Ferrara, Marta Luigina Mangini, Sergio Picascia and Ambra Stefanello

The Rise and Fall of Dependency Parsing in Dante Alighieri's Divine Comedy
Claudia Corbetta, Marco Passarotti and Giovanni Moretti

Unsupervised Authorship Attribution for Medieval Latin Using Transformer-Based Embeddings
Loic De Langhe, Orphee De Clercq and Veronique Hoste

"To Have the 'Million' Readers Yet": Building a Digitally Enhanced Edition of the Bilingual Irish-English Newspaper an Gaodhal (1881-1898)
Oksana Dereza, Deirdre Ní Chonghaile and Nicholas Wolf

Introducing PaVeDa – Pavia Verbs Database: Valency Patterns and Pattern Comparison in Ancient Indo-European Languages
Silvia Luraghi, Alessio Palmero Aprosio, Chiara Zanchi and Martina Giuliani

Saturday, May 25, 2024 (continued)

+ ***EvaLatin***

Saturday, May 25, 2024 (continued)

Behr at EvaLatin 2024: Latin Dependency Parsing Using Historical Sentence Embeddings
Rufus Behr

KU Leuven / Brepols-CTLO at EvaLatin 2024: Span Extraction Approaches for Latin Dependency Parsing
Wouter Mercelis

ÚFAL LatinPipe at EvaLatin 2024: Morphosyntactic Analysis of Latin
Milan Straka, Jana Straková and Federica Gamba

Nostra Domina at EvaLatin 2024: Improving Latin Polarity Detection through Data Augmentation
Stephen Bothwell, Abigail Swenor and David Chiang

TartuNLP at EvaLatin 2024: Emotion Polarity Detection
Aleksei Dorkin and Kairit Sirts

+ ***EvaHan***

Overview of EvaHan2024: The First International Evaluation on Ancient Chinese Sentence Segmentation and Punctuation
Bin Li, Bolin Chang, Zhixing Xu, Minxuan Feng, Chao Xu, Weiguang QU, Si Shen and Dongbo Wang

Two Sequence Labeling Approaches to Sentence Segmentation and Punctuation Prediction for Classic Chinese Texts
Xuebin Wang and Zhenghua Li

Ancient Chinese Sentence Segmentation and Punctuation on Xunzi LLM
Shitu Huo and Wenhui Chen

Sentence Segmentation and Sentence Punctuation Based on XunziALLM
Zihong Chen

Sentence Segmentation and Punctuation for Ancient Books Based on Supervised In-context Training
Shiquan Wang, Weiwei Fu, Mengxiang Li, Zhongjiang He, Yongxiang Li, Ruiyu Fang, Li Guan and Shuangyong Song

SPEADO: Segmentation and Punctuation for Ancient Chinese Texts via Example Augmentation and Decoding Optimization
Tian Xia, Kai Yu, Qianrong Yu and Xinran Peng

Ancient Chinese Punctuation via In-Context Learning
Jie Huang

Goidelex: A Lexical Resource for Old Irish

Cormac Anderson[†], **Sacha Beniamine**[†], **Theodorus Fransen**[‡]
[†]University of Surrey, Guildford, United Kingdom
[‡]Università Cattolica del Sacro Cuore, Milan, Italy
{cormac.anderson, s.beniamine}@surrey.ac.uk, theodorus.fransen@unicatt.it

Abstract

We introduce Goidelex, a new lexical database resource for Old Irish. Goidelex is an openly accessible relational database in CSV format, linked by formal relationships. The launch version documents 695 headwords with extensive linguistic annotations, including orthographic forms using a normalised orthography, automatically generated phonemic transcriptions, and information about morphosyntactic features, such as gender, inflectional class, etc. Metadata in JSON format, following the Frictionless standard, provides detailed descriptions of the tables and dataset. The database is designed to be fully compatible with the Paralex and CLDF standards and is interoperable with existing lexical resources for Old Irish such as CorPH and DIL. It is suited to both qualitative and quantitative investigation into Old Irish morphology and lexicon, as well as to comparative research. This paper outlines the creation process, rationale, and resulting structure of the database.

Keywords: Old Irish, morphology, lexicon, inflection

1. Introduction

We present Goidelex,[1] a new lexical database of Old Irish[2] which draws on and adds to existing digital resources for the language. The launch version of the database documents 695 headwords, in both orthographic forms and phonemic transcription and with extensive linguistic annotation. It is structured and formatted as a set of CSV files and is designed to be forward compatible with the Paralex and CLDF standards (Beniamine et al., 2023; Forkel et al., 2018). While the launch dataset contains only nominal lexemes, the database has been designed with a view to adding also other parts of speech in the near future. As a standalone resource, Goidelex is suited for both qualitative and quantitative investigation of the Old Irish lexicon. Moreover, it links between several existing resources: the electronic Dictionary of the Irish Language (DIL: Toner et al. (2013-present)), the Corpus PalaeoHibernicum (CorPH: Stifter et al. (2021)) and the Würzburg glosses (Kavanagh and Wodtko, 2001), facilitating research on Old Irish phonology and morphology.

In recent times, computational methods have increasingly come to be used to investigate various aspects of linguistic typology and evolution. However, these methods require well-structured machine-readable data, which is most often available only for well-resourced, literary languages with lots of speakers (Dahl, 2015; Malouf et al., 2020; Bird, 2022). This unbalanced sampling makes it especially important to develop new datasets, or uplift existing ones, for lesser-studied languages. More datasets for minoritised languages, which frequently document rare linguistic features (Mithun, 2007), will bring more precision to our measurements of the synchronic distribution of linguistic features. Better data availability for historical languages, especially where comparable data for cognate or daughter languages is also available, will improve our understanding of the dynamics underlying language evolution.

The Goidelic languages are an obvious target case for improved data development. They comprise a well-defined language cluster exhibiting features diverging from the areal and cross-linguistic norm at all levels of linguistic structure. All surviving Goidelic languages – Irish[3], Manx[4], and Scottish Gaelic[5] – are minoritised. While their development from Old Irish (600-900CE) is well-documented in the textual record, it has not been comprehensively described.

Old Irish itself is the earliest Celtic language for which attestation is copious enough to allow for a full grammatical description. It is noticeably divergent from related Indo-European languages. Syntactically, like other Insular Celtic languages, it has dominant verb-initial word order (Thurneysen, 1946). Morphologically, it shows extremely complex patterns of verbal inflection, even by the standards of older Indo-European languages (McCone, 1987). Phonologically, it has a large consonant system and has been described as having a vertical vowel system (Anderson, 2016). Given this linguistic profile, good computational resources for the language are an urgent desideratum.

Our contributions in this paper are the following:

[1] DOI: 10.5281/zenodo.10898227; repository: https://github.com/cormacanderson/Goidelex

[2] ISO 639-3 code `sga`; Glottocode `oldi1246`

[3] ISO 639-3 code `gle`; Glottocode `iris1253`

[4] ISO 639-3 code `glv`; Glottocode `manx1243`

[5] ISO 639-3 code `gla`; Glottocode `scot1245`

- A lexical resource for Old Irish, interoperable with existing resources (CorPH, DIL), and providing a unified, standardised representation of lexemes and structured groupings into lexemes and flexemes (Fradin and Kerleroux, 2003; Thornton, 2018; Pellegrini, 2023).

- Normalised orthography, providing a single identifier for orthographic variants of a single lexeme.

- Generated phonological forms, facilitating morphological and phonological research.

- Detailed morphosyntactic and morphonological annotation, including part of speech, inflectional class, gender, propensity to syncope, etc.

- Information on etymology and derivational family for each lexeme.

- A manually curated set of rules for grapheme-to-phoneme conversion, starting from the normalised orthography.

- Progress towards the digitisation of the Würzburg glosses.

2. Previous work

The main digital lexical resources available for Old Irish are the electronic Dictionary of the Irish Language (DIL: Toner et al. (2013-present) and the Corpus PalaeoHibernicum (CorPH: Stifter et al. (2021)).

DIL is a longstanding dictionary resource, originally available in print format, but in recent times also online. Its lexical coverage of the language is comprehensive, but search and filter functions are quite rudimentary and the orthography of headwords inconsistent, making the assembly of examples for linguistic research very difficult. Furthermore, examples are only sometimes annotated for morphosyntactic features, making it difficult to use DIL for morphological investigation. Further, DIL has not been digitised in a way that makes it easy to extract the data computationally.

CorPH attempts to resolve these problems. While it operates over a smaller corpus than DIL, it is far more thorough in terms of morphosyntactic annotation, making it much more useful for morphological research. It still suffers from certain limitations, however, which create difficulties in terms of aggregating data. In particular, the orthography of headwords is not fully standardised, so lexemes with the same phonological profile may be spelled differently, and there are occasional duplicate entries where a single lexeme has two separate entries with differing orthography. It also does not include the Würzburg glosses, one of the most important contemporary sources for Old Irish.

At present, no digital lexical resource exists for the Würzburg glosses. While a digital edition of the text (Doyle, 2018) and a UD treebank containing a small selection of glosses are available (Doyle, 2023), these resources do not provide fine-grained phonological or morphological annotation. The most comprehensive source, and therefore most suitable for our purposes, remains the printed lexicon (Kavanagh and Wodtko, 2001).

The limitations of these existing resources create difficulties both for end-users and for linguistics researchers. For the end-user, it is difficult to find lexemes, as there is no orthographic normalisation, meaning one must try variant spellings until one finds the lexeme one is looking for. Compounding this, the search and filter capabilities in DIL are very limited, although CorPH is considerably better in this respect. For the researcher, orthographic inconsistency makes it very difficult to assemble examples for linguistic comparison and has impeded the development of standard NLP tools such as grapheme-to-phoneme conversion.

Goidelex aims to address these limitations. It provides a consistent and standardised lexical resource that will be useful as a lexical resource for both studies in Old Irish phonology, morphology and lexicon, and wider comparative linguistics research. Beyond its standalone value, it has broader function as a basis from which to produce other lexical resources, such as inflected lexicons for morphology (e.g. the Paralex datasets, Beniamine et al. 2023), concept and cognacy-coded word lists for historical linguistics (e.g. the CLDF datasets, Forkel et al. 2018), and, following Mambrini and Passarotti (2023), a lemma collection modelled as a knowledge graph according to Ontolex, the W3C de-facto standard for lexical information in the Linked Data paradigm (McCrae et al., 2017).

Goidelex focuses on the lexicon of the Würzburg glosses in the first instance, as this material is not available through CorPH. While the initial dataset has only nominal lexemes, the database will be expanded to include other parts of speech in the near future.

3. Design principles

A first problem to be confronted when developing a lexical resource for Old Irish is the ambiguous and inconsistent nature of the language's orthography. As mentioned in § 2, inconsistent spelling of headwords makes it difficult to search a resource for a given lexeme or to filter lexemes to draw up a list of examples for research. This inconsistency also hampers the development of NLP tools, such as grapheme-to-phoneme conversion. Our solution to this problem in Goidelex was to use a normalised orthography (see § 3.1).

Beyond orthography, the Old Irish lexicon exhibits considerable variation at all levels of linguistic structure. In some cases, the same lexeme shows different phonological forms across surviving corpora. In others, there are differences in inflection, be it in morphonological behaviour, such as the occurrence of syncope in certain forms, or in morphosyntactic properties such as gender or inflectional class. We attempt to capture this variation by a principled distinction between lexemes and flexemes (§ 3.2).

3.1. Normalised orthography

A key innovation of Goidelex is the use of a normalised orthography for citation forms. This has a number of advantages. First, existing sources frequently differ in the spelling of the citation forms they use for any given lexeme, which makes it difficult to identify lexemes within and across sources. The normalised orthography provides a principled representation that makes it easier for users of the database to find lexemes. Second, it provides a human readable form that serves as an identifier to link data from different corpora. As such, it is a secure basis for lemmatisation, reducing considerably the risk of duplicate headwords (as occur occasionally in Stifter et al., 2021). Third, and most critically, it constitutes a standardised starting point for grapheme-to-phoneme conversion (§ 4.3).

We follow the normalised orthography proposed by Fransen et al. (2023), which adheres to six basic principles: comprehensiveness, clarity, neutrality, redundancy, fidelity, and conventionality. It is intended to represent all possible forms in Old Irish. Each normalised orthographic form aims to correspond to a single phonological form, while for each phonological form there is a single, obvious, orthographic representation. The normalised orthography remains as neutral as possible with respect to different phonological analyses and makes ample use of redundancy in cases of uncertainty. It aims to be as faithful as possible to genuine Old Irish spelling and to existing scholarly conventions.

3.2. Lexemes and flexemes

In Goidelex, we take lexemes to be defined by a shared meaning and a single part of speech. This means, for example, that deadjectival nouns are to be listed separately from the adjectives from which they derive, and denominal verbs are to be listed separately from the nouns from which they are formed. Derivational relationships between lexemes are captured by the notion of derivational families (§ 5.2).

However, a single lexeme sometimes leads to multiple distinct inflectional paradigms, due to variation in its phonology, morphonology, or morphosyntactic behaviour. To capture this variation, we use the notion of *flexeme* (Fradin and Kerleroux, 2003; Thornton, 2018; Pellegrini, 2023). In this approach, each inflectional variant of a lexeme, differing in terms of phonology, morphonology, or morphosyntax, is analysed as a separate flexeme. Thus, a single lexeme may map to multiple flexemes.

Some examples can serve to illustrate this. The noun *muinter* ‘family, household’ sometimes appears as *muinter* and sometimes as *muntar*. These different spellings reflect a phonological difference: the cluster is palatalised /$n^{j}t^{j}$/ in the first instance and labiovelarised /$n^{w}t^{w}$/ in the second. On this basis, we have two separate flexemes, both linked to the same lexeme entry. A futher example is provided by the noun *fius*, which does not vary in terms of its phonology, but which varies with respect to morphosyntactic category. Sometimes it is inflected as a neuter u-stem, sometimes as a masculine u-stem, and sometimes as a neuter o-stem. We thus set up three different flexemes corresponding to this single lexeme.

Identifying flexemes required detailed manual study of the attested forms of each lexeme appearing in the Würzburg and CorPH datasets. A total of 107 out of 574 lexemes showed variation either in terms of their phonology, their morphonological patterning, their gender, or their inflectional class. In total, there are 695 flexemes in the Goidelex launch dataset, corresponding to 574 lexemes.

4. Building the database

We produced the database in three steps. First, we manually entered lemmata from the Würzburg glosses into the `Lexeme` and `Flexeme` tables (§ 4.1). Then, we merged lexemes with CorPH lemmata in a semi-automatic fashion (§ 4.2). Finally, we carried out automatic grapheme-to-phoneme conversion using customised rules (§ 4.3). Further tables were input manually.

4.1. Würzburg lemmata entry

The first stage of data collection involved manually entering nouns from the Würzburg glosses. This corpus was chosen as it is by far the largest and most important corpus of Old Irish for which no digital lexical resource was available. All nouns with more than one attestation in the lexicon of the Würzburg glosses (Kavanagh and Wodtko, 2001) were included, amounting initially to a total of 574

nouns.

This yielded a list of orthographic headwords, to which we manually added a detailed part of speech, gender, inflection class, gloss, derivational family annotation, and url references to entries in the electronic Dictionary of the Irish language (Toner et al., 2013-present). Detailed study of the Würzburg glosses was necessary in order to identify orthographic, morphological or phonological variation and conduct a preliminary analysis of entries into lexemes and flexemes. To facilitate bridging across resources as well as phonological transcription, we manually transcribed each lemma into the normalised orthography proposed by Fransen et al. (2023).

4.2. Merging with CorPH lemmata

Lexical entries were then aligned with corresponding data in CorPH (Stifter et al., 2021). First, we manually annotated each lexeme with the corresponding headwords in CorPH. Then, leveraging these headwords, we automatically extracted from CorPH lemma ID numbers, full meaning definitions (more complete than our short glosses), and etymological information. We flagged potential problems and manually corrected all cases in which there were mismatches between our annotations and those found in CorPH. In certain instances, this involved adding also new flexemes to capture variation in the CorPH dataset that is not present in the Würzburg corpus.

4.3. Grapheme-to-phoneme conversion

We then generated phonological forms from the lexemes in normalised orthography using handmade rules. As well as being suitable for cross-linguistic comparison, phonological forms are a principled basis from which to develop new tools and resources for Old Irish.

We write phonological forms according to the phonological system set out in Anderson (2016). In order to convert normalised orthographic forms to this representation we used the Epitran software (Mortensen et al., 2018) to process our grapheme-to-phoneme rules. Epitran proceeds in three steps, each applied independently on input forms (here citation forms in normalised orthography):

1. **Preprocessing**: a first set of ordered rules.
2. **Mapping**: a set of non-contextual mappings.
3. **Postprocessing**: a second set of ordered rules.

Epitran rules are written according to a custom syntax that resembles traditional phonological rules, employing variables and regular expressions. We devised our own set of rules and grouped them into numbered blocks to facilitate readability, identification of errors, and validation.

5. Structure of the database

The database is structured as a set of CSV files, linked by foreign key relations (see Figure 1). Since no standard existed for the specific type of data in question here, we chose formats and structures compatible with related standards. In particular, Goidelex is designed to be easily extended (see § 6) into datasets fitting either the Paralex standard for inflected lexicons (Beniamine et al., 2023) or the Cross-Linguistic Data Format standard suitable for cognate-coded lexical data (Forkel et al., 2018).

5.1. Lexemes table

The `Lexemes` table (Table 1.a) identifies individual lexemes and links these to other Old Irish resources. Lexemes are identified by a unique identifier, as well as by a human readable citation form written in the normalised orthography developed for this project. Entries in the `Lexemes` table are linked to two online Old Irish resources (Stifter et al., 2021; Toner et al., 2013-present) and to the Würzburg dictionary (Kavanagh and Wodtko, 2001).

Information about lexemes was mostly manually annotated by the authors, but in some cases was drawn from CorPH. Each lexeme is defined as belonging to a single part of speech, meaning, for example, that adjectives have separate lexical entries to deadjectival nouns formed from them, while verbal nouns are listed separately to the verbs to which they are associated. However, related lexemes are aggregated into derivational families (§ 5.2).

- **lexeme_id**: The primary key for this table, it identifies the entire row and acts as a foreign key in the `Flexemes` table (§ 5.3).
- **derivational_families**: Foreign key identifier(s) from the `Derivational families` table (§ 5.2), separated by semicolons where more than one entry.
- **label**: Human readable citation form for the lexeme in normalised orthography.
- **CorPH_ids**: The identification number(s) of the corresponding lexeme in the CorPH database.
- **CorPH_labels**: The citation form(s) of the corresponding lexeme in the CorPH database.

(a) Lexemes table

lexeme_id	derivational _families	label	CorPH_ids	CorPH_labels	CorPH_meaning	Wb_label	DIL_URL	gloss	POS
lex-apstal-56	55	apstal	3389	apstal	apostle	apstal, abstal	http://dil.ie/3887	apostle	noun
lex-apstalach-57	55	apstalach	3390	apstalacht	apostleship, apostolate	apstalacht	http://dil.ie/3889	apostolate	noun
lex-brithem-98	92	brithem	3573	brithem	judge	brithem	http://dil.ie/6699	judge	noun
lex-firinne-280	262	fírinne	4620	fírinne	truth; justice, righ[...]	fírinne	http://dil.ie/22203	righteousness, truth	noun
lex-fius-282	54	fius	4627	fius	the act of finding o[...]	fiuss, fius, fis	http://dil.ie/22221	knowledge	verbal_noun
lex-muinter-429	400	muinter	2187	muinter	family, household, f[...]	muntar	http://dil.ie/32754	community	noun
lex-talam-525	482	talam	5880	talam	earth, world; ground[...]	talam	http://dil.ie/39932	earth	noun

(b) Flexemes table

flexeme_id	label	lexeme	texts	etymology	inherent_properties	phonological_form
apstal-56	apstal	lex-apstal-56			sync_none;alt_none;gen_masc;stem_o;num_all	ˈØˠapˠsˠtˠalˠ
apstalach-57	apstalach	lex-apstalach-57			sync_none;alt_none;gen_fem;stem_ā;num_all	ˈØˠapˠsˠtˠalˠaxˠ
brithem-98	brithem	lex-brithem-98	79;6;12	denominative (+ agent suffix -em): < breth	sync_none;alt_none;gen_masc;stem_n;num_all	ˈbʲrʲəθʲaμˠ
breithem-98.1	breithem	lex-brithem-98	5	denominative (+ agent suffix -em): < breth	sync_none;alt_none;gen_masc;stem_n;num_all	ˈbʲrʲaθʲaμˠ
firinne-280	fírinne	lex-firinne-280		denominative (abstract): < fírión/fírián/fíré[...]	sync_vf;alt_none;gen_fem;stem_iā;num_all	ˈφʲəØʲrʲəɴʲaØʲ
fius-282	fius	lex-fius-282		*u̯id-tu-; vn. of ro·fitir	sync_none;alt_none;gen_neut;stem_u;num_all	ˈφʲəsʷ
fius-282.1	fius	lex-fius-282		*u̯id-tu-; vn. of ro·fitir	sync_none;alt_none;gen_masc;stem_u;num_all	ˈφʲəsʷ
fius-282.2	fius	lex-fius-282		*u̯id-tu-; vn. of ro·fitir	sync_none;alt_none;gen_neut;stem_o;num_all	ˈφʲəsʷ
muinter-429	muinter	lex-muinter-429	79;70;5;1	<Lat. monasterium? via British?	sync_none;alt_none;gen_fem;stem_ā;num_all	ˈmʷənʲtʲarˠ
muntar-429.1	muntar	lex-muinter-429	6;7	<Lat. monasterium? via British?	sync_none;alt_none;gen_fem;stem_ā;num_all	ˈmʷənʷtʷarˠ
talam-525	talam	lex-talam-525		*telamon-	sync;alt_none;gen_masc;stem_n;num_all	ˈtˠalˠaμˠ

(c) Inherent properties table

properties_id	label	comment	domain	type
alt_none	No morphonological alternations	This flexeme does not show morphonological alternations	morphonology	alternation
sync_none	No syncope	The vowel of the final syllable of this flexeme is not liable to syncope	morphonology	syncope
sync_vf	Vowel final	This flexeme ends in a vowel, which is liable to be lost	morphonology	syncope
sync	Syncope	The vowel of the final syllable of this flexeme is liable to syncope	morphonology	syncope
gen_masc	Masculine	This flexeme has masculine gender	morphosyntax	gender
gen_fem	Feminine	This flexeme has feminine gender	morphosyntax	gender
gen_neut	Neuter	This flexeme has neuter gender	morphosyntax	gender
stem_o	o-stem noun	This flexeme is inflected as an o-stem	morphosyntax	nominal_stem
stem_ā	ā-stem noun	This flexeme is inflected as an ā-stem	morphosyntax	nominal_stem
stem_n	n-stem noun	This flexeme is inflected as an n-stem	morphosyntax	nominal_stem
stem_iā	iā-stem	This flexeme is inflected as an iā-stem	morphosyntax	nominal_stem
stem_u	u-stem noun	This flexeme is inflected as a u-stem	morphosyntax	nominal_stem
num_all	No number restriction	This flexeme is inflected for all numbers	morphosyntax	number_restriction

Table 1: Excerpts from the Lexeme, Flexeme and Inherent Properties tables (long cells are truncated).

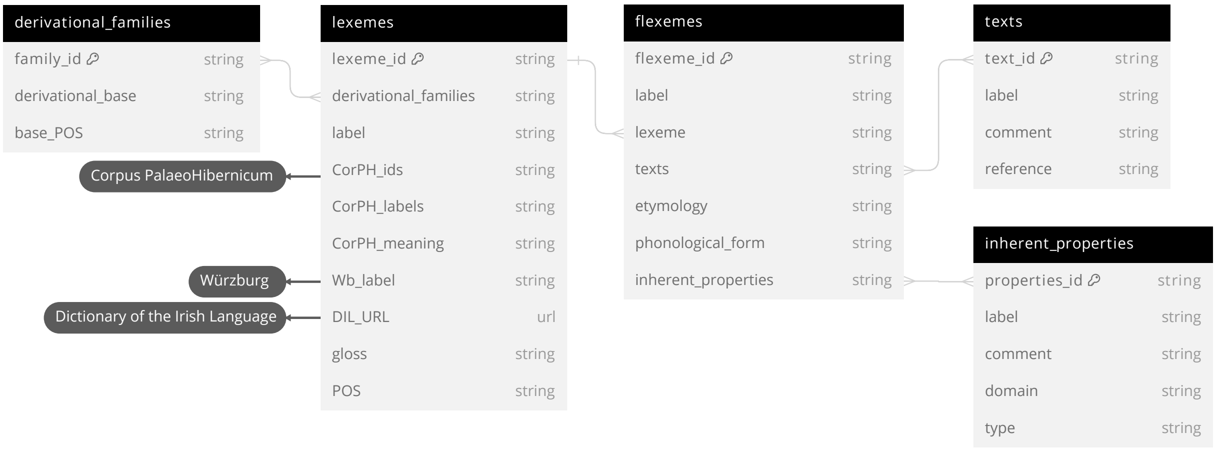


Figure 1: Database schema. External resources are indicated as rounded boxes.

- **CorPH_meaning**: A full meaning description for the lexeme extracted from the CorPH database.

- **DIL_url**: The url of the corresponding lexeme in the online Dictionary of the Irish Language (DIL).

- **Wb_label**: The citation form of the corresponding lexeme in the lexicon of the Würzburg glosses (Kavanagh and Wodtko, 2001).

- **gloss**: A short description of the meaning of the lexeme.

- **POS**: The part of speech of the lexeme.

5.2. Derivational families table

family_id	**derivational_base**	**base_POS**
55	apstal	noun
92	beirid	verb
262	fíor	adjective
54	ro·fitir	verb
400	muinter	noun
482	talam	noun

Table 2: A few rows from the Derivational families table

The `Derivational families` table (Table 2) links together lexemes that stand in a derivational relationship (including compounds). In some cases, this does not involve a change in part of speech, e.g. *apstal* 'apostle' and *apstalacht* 'apostolate' are both nouns. In other cases, linked lexemes can have different parts of speech, e.g. *fíor* 'true' is an adjective, whereas *fírinne* 'truth' is a noun.

- **family_id**: The primary key for this table, it identifies the entire row and acts as a foreign key in the `Lexemes` table (§ 5.1).

- **derivational_base**: The citation form in normalised orthography for the derivational base of a lexeme.[6]

- **base_POS**: The part of speech of the derivational base.

5.3. Flexemes table

A single lexeme can sometimes show phonological, morphonological or morphosyntactic variation. We thus define flexemes as finely-grained variants of lexemes (Fradin and Kerleroux, 2003; Thornton, 2018; Pellegrini, 2023), each belonging to a single inflectional microclass (Dressler, 2002; Beniamine et al., 2018). In Goidelex, each flexeme has a unique identifier and a label in normalised orthography (Table 1.b). It is linked by foreign keys to its parent lexeme and we provide further information regarding textual distribution, inflection class, etymology, and any morphonological particularities that are not predictable from its orthographic form.

- **flexeme_id**: The primary key for this table, this identifies the entire row. Derived resources are expected to refer to these identifiers.

- **label**: A human readable label for the flexeme in normalised orthography.

- **lexeme**: A foreign key identifying the parent lexeme in the `Lexemes` table (§ 5.1).

[6] At this point, we treat the citation form of the associated verb as the derivational base of a verbal noun. This idealisation will facilitate expansion of the dataset to also include verbal forms.

- **texts**: The set of available texts in which this variant occurs, given as text codes separated by a semicolon. Text codes are foreign key identifiers from the `Texts` table (§ 5.5).

- **etymology**: The etymology of the flexeme, drawn from CorPH.

- **inherent_properties**: A set of foreign keys, separated by semicolon, identifying non-predictable morphonological or morphosyntactic information about the lexeme in the `Inherent properties` table (§ 5.4).

- **phonological_form**: A phonological form generated by grapheme-to-phoneme conversion.

Full normalisation of `inherent_properties` would have required an intermediate table mapping identifiers from the flexeme and inherent properties tables. However, such a (very long) table would be nearly unusable to users reading the database in spreadsheet software, or who may not be able to perform database joins. Conversely, setting the properties in wide format (as is often the preference for qualitative research), with columns for each type of property, would have made the table very specific to nominal entries, and would lead to a large increase in columns for verbal entries. Our choice here is instead a compromise between normalisation and ease of use: by keeping the long form, we can fully describe each property in the relevant table (§ 5.4), while ensuring that these properties can be read directly from the relevant rows of the `Flexemes` table, which is more intuitive to less technical users. This comes at the cost of adding cell-internal separators (here, semicolons), a choice we resorted to also for a few other columns, such as `flexemes.texts`, `lexemes.CorPH_ids`, `lexemes.CorPH_labels`.

5.4. Inherent properties table

As described in § 3.2, some flexemes have inherent properties (Table 1.c) that are not predictable from their normalised orthographic form. These include morphonological properties as well as inherent morphosyntactic information such as gender and inflection class.

A common example of a non-predictable morphonological property of a flexeme, which is annotated in this table, is propensity to syncope. For example, *talam* ‘land’ and *brithem* ‘judge’ are both masculine n-stem nouns. However, *talam* has the genitive singular form *talman*, with syncope of the second syllable, while *brithem* has the genitive singular form *britheman*, without syncope.

Morphonosytactic properties annotated here include gender (masculine, neuter, and feminine in Old Irish), and inflectional class. As with the morphonological properties, these morphosyntactic properties are not predictable from the orthographic form, so must be annotated for each individual flexeme.

The `Inherent properties` table lists all valid codes for these properties. The launch version of Goidelex has only nouns, so currently only properties relevant to nouns need to be annotated. Further rows will be added here in future as we expand the database to include also other parts of speech.

- **properties_id**: The primary key of this table and a foreign key in the `Flexemes` table (§ 5.3).

- **label**: A human readable label identifying the phonological or morphonological property to which the identifier refers.

- **comment**: The text description of the property described.

- **domain**: Properties pertain to different linguistic domains. The domains in use are `morphonology` and `morphosyntax`.

- **type**: Properties can be logically grouped. This field assigns a type grouping to each class identified. The types in use are: `gender`, `stem_class`, `alternation`, `syncope`, `number_restriction`.

5.5. Texts table

Some flexemes occur in one set of texts, while others occur in other texts within the Old Irish corpus. This table (Table 3) provides explicit information regarding the texts in which a particular variant is found.

- **text_id**: The primary key of this table and a foreign key in the `Flexemes` table (§ 5.3). The text IDs are the same as those used in CorPH, with some extension to include texts (predominantly the Würzburg glosses) which do not occur in that dataset.

- **label**: A human readable label identifying the text.

- **comment**: A text description of the text in question.

- **reference**: A bibliographic reference for this text, in most cases also taken from CorPH.

text_id	label	Comment	reference
1	Annals of Ulster		Mac Airt and Mac Niocaill 1983
2	Vita Columbae		Anderson and Anderson 1961; Thes. II, 272–280
3	Baile Chuinn		Murray and Bhreathnach 2005
4	Disciples and Relatives of Columba		Anderson and Anderson 1961;Thes. II, 281
5	Poems of Blathmac		Barrett 2018; Carney 1964

Table 3: A few rows from the texts table

6. Conclusion and future work

Goidelex has been designed to act as a central lexical resource for Old Irish. It aligns data from multiple sources, provides central identifiers and normalised representations, as well as very detailed phonological and morphological annotation. The database makes this information accessible for a wide range of qualitative and quantitative purposes.

Many open questions about Old Irish phonology and morphology can be addressed using the database. The structured nature of the Goidelex data makes it easy to collect examples for investigation from corpora, something which is difficult or impossible with existing resources. Consistent annotation of phonological and morphosyntactic properties opens up numerous possibilities for research into Old Irish phonology and morphology, while the grouping of lexemes into derivational families facilitates studies of word formation.

As a central resource, Goidelex lends itself to extensions as separate datasets. In particular, we envision three types of derived datasets: inflected lexicons, cross-linguistic cognacy datasets, and a linked lemma bank.

Goidelex constitutes a sound basis from which to develop an inflected lexicon of the Old Irish noun, compatible with the Paralex standard (Beniamine et al., 2023). Indeed, the surface inflectional paradigms of each flexeme are fully predictable from the phonological transcriptions and the morphonological and morphosyntactic annotations documented in Goidelex. This fine-grained information can serve as the input for finite-state transducers in order to generate full inflected paradigms. Currently, no such resource exists (the Old Irish Unimorph dataset (Batsuren et al., 2022) counts 50 verbs, only in orthography, and with no nominal paradigms).

Goidelex is also meant to serve as a basis for developing cross-linguistic comparative cognacy data for the Goidelic languages. There already exists a bridge (Scannell, 2018), which links entries between the the most important dictionaries of Old Irish (DIL: Toner et al., 2013-present) and Modern Irish (Dónaill, 1977) and similar work is under way to link Modern Irish to other modern Goidelic languages. The design of Goidelex, being compatible with the CLDF standard (Forkel et al., 2018), facilitates efforts to align cognate data to other languages.

Finally, ongoing work (Fransen et al., 2024), inspired by similar efforts for Latin (Mambrini and Passarotti, 2023), uses a subset of Goidelex to create a lemma bank for Old Irish within the Linked Data paradigm.

7. Ethical statement

To the best of our knowledge there are no ethical concerns pertaining to this resource.

8. Acknowledgements

Cormac Anderson is funded by a British Academy Grant (GP GP300169) while Sacha Beniamine is funded by a Leverhulme Early Career Fellowship (ECF-2022-286). Theodorus Fransen has received funding from the European Union's Horizon Europe scientific research initiative under the Marie Skłodowska-Curie Actions (MSCA), grant agreement No 101106220 (MOLOR – Morphologically Linked Old Irish Resource).

9. Bibliographical references

Cormac Anderson. 2016. *Consonant colour and vocalism in the history of Irish*. Ph.D. thesis, Adam Mickiewicz University, Poznań.

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieraś, Gábor Bella, Brian Leonard, Garrett Nicolai, Kyle Gorman, Yustinus Ghanggo Ate, Maria Ryskina, Sabrina Mielke, Elena Budianskaya, Charbel El-Khaissi, Tiago Pimentel, Michael Gasser, William Abbott Lane, Mohit Raj, Matt Coler, Jaime Rafael Montoya Samame, Delio Siticonatzi Camaiteri, Esaú Zumaeta Rojas, Didier López Francis, Arturo Oncevay, Juan López Bautista, Gema Celeste Silva Villegas, Lucas Torroba Hennigen, Adam Ek, David

Guriel, Peter Dirix, Jean-Philippe Bernardy, Andrey Scherbakov, Aziyana Bayyr-ool, Antonios Anastasopoulos, Roberto Zariquiey, Karina Sheifer, Sofya Ganieva, Hilaria Cruz, Ritván Karahóğa, Stella Markantonatou, George Pavlidis, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Candy Angulo, Jatayu Baxi, Andrew Krizhanovsky, Natalia Krizhanovskaya, Elizabeth Salesky, Clara Vania, Sardana Ivanova, Jennifer White, Rowan Hall Maudslay, Josef Valvoda, Ran Zmigrod, Paula Czarnowska, Irene Nikkarinen, Aelita Salchak, Brijesh Bhatt, Christopher Straughn, Zoey Liu, Jonathan North Washington, Yuval Pinter, Duygu Ataman, Marcin Wolinski, Totok Suhardijanto, Anna Yablonskaya, Niklas Stoehr, Hossep Dolatian, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Aryaman Arora, Richard J. Hatcher, Ritesh Kumar, Jeremiah Young, Daria Rodionova, Anastasia Yemelina, Taras Andrushko, Igor Marchenko, Polina Mashkovtseva, Alexandra Serova, Emily Prud'hommeaux, Maria Nepomniashchaya, Fausto Giunchiglia, Eleanor Chodroff, Mans Hulden, Miikka Silfverberg, Arya D. McCarthy, David Yarowsky, Ryan Cotterell, Reut Tsarfaty, and Ekaterina Vylomova. 2022. UniMorph 4.0: Universal Morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 840–855, Marseille, France. European Language Resources Association.

Sacha Beniamine, Cormac Anderson, Mae Carroll, Matías Guzmán Naranjo, Borja Herce, Matteo Pellegrini, Erich Round, Helen Sims-Williams, and Tiago Tresoldi. 2023. Paralex: a dear standard for rich lexicons of inflected forms. In *International Symposium of Morphology*. https://www.paralex-standard.org.

Sacha Beniamine, Olivier Bonami, and Benoît Sagot. 2018. Inferring inflection classes with description length. *Journal of Language Modelling*, 5(3):465–525.

Steven Bird. 2022. Local languages, third spaces, and other high-resource scenarios. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 7817–7829, Dublin. Association for Computational Linguistics.

Östen Dahl. 2015. How WEIRD are WALS languages? Paper presented at the Diversity linguistics - retrospect and prospect conference, Max Planck Institute for Evolutionary Anthropology, May 1-3, 2015, Leipzig.

Wolfgang Dressler. 2002. Latin inflection classes. In A. Machtelt Bolkestein, Caroline H.M. Kroon, Harm Pinkster, and Rodie Risselada H. Wim Remmelink, editors, *Theory and description in Latin linguistics: Selected Papers from the XIth International Colloquium on Latin Linguistics*, pages 91–110. Brill, Leiden.

Robert Forkel, Johann-Mattis List, Simon J. Greenhill, Christoph Rzymski, Sebastian Bank, Michael Cysouw, Harald Hammarström, Martin Haspelmath, Gereon A. Kaiping, and Russell D. Gray. 2018. Cross-Linguistic Data Formats, advancing data sharing and reuse in comparative linguistics. *Scientific Data*, 5(180205).

Bernard Fradin and Françoise Kerleroux. 2003. Troubles with lexemes. In Geert Booij, Janet DeCesaris, Angela Ralli, and Sergio Scalise, editors, *Selected papers from the third Mediterranean Morphology Meeting*, pages 177–196. IULA – Universitat Pompeu Fabra.

Theodorus Fransen, Cormac Anderson, and Sacha Beniamine. 2023. Towards a normalised orthography for Old Irish. Paper at *36th Irish Congress of Medievalists*, Dublin, 22–23 June 2023.

Theodorus Fransen, Cormac Anderson, Sacha Beniamine, and Marco Passarotti. 2024. The MOLOR Lemma Bank: A new LLOD resource for Old Irish. Paper accepted to the *9th Workshop on Linked Data in Linguistics (LDL-2024)* at LREC-Coling 2024.

Séamus Kavanagh and Dagmar S. Wodtko. 2001. *A lexicon of the Old Irish glosses in the Würzburg manuscript of the epistles of St. Paul*. Verlag der Österreichischen Akademie der Wissenschaften, Vienna.

R. Malouf, F. Ackerman, and A. Semenuks. 2020. Lexical databases for computational analyses: A linguistic perspective. *Society for Computation in Linguistics*, 3(1):297–307.

Francesco Mambrini and Marco Carlo Passarotti. 2023. The LiLa Lemma Bank: A Knowledge Base of Latin canonical forms. *Journal of Open Humanities Data*, 9(28):1–5.

Kim McCone. 1987. *The Early Irish verb*. An Sagart, Maynooth.

John P. McCrae, Julia Bosque-Gil, Jorge Gracia, Paul Buitelaar, and Philipp Cimiano. 2017. The OntoLex-Lemon Model: Development and Applications. In *Electronic lexicography in the 21st century. Proceedings of eLex 2017 conference*, pages 587–597, Brno, Czech Republic. Lexical Computing CZ s.r.o.

Marianne Mithun. 2007. Linguistics in the face of language endangerment. In Leo W. Wetzels, editor, *Language endangerment and endangered languages: Linguistic and anthropological studies with special emphasis on the languages and cultures of the Andean-Amazonian border area (Indigenous Languages of Latin America (ILLA)*, volume 5 of *Publications of the Research School of Asian, African, and Amerindian Studies (CNWS) 154)*, pages 15–35. Research School CNWS, Leiden University, Leiden.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Matteo Pellegrini. 2023. Flexemes in theory and in practice. *Morphology*, 33:361–395.

Anna M. Thornton. 2018. Troubles with flexemes. In Oliver Bonami, Gilles Boyé, Hélène Firaudo, and Fiammetta Namer, editors, *The lexeme in descriptive and theoretical morphology*, pages 202–321. Language Science Press, Berlin.

Rudolf Thurneysen. 1946. *A Grammar of Old Irish*. Dublin Institute of Advanced Studies, Dublin. Translated by D. A. Binchy and Osborn Bergin.

10. Language resource references

Adrian Doyle. 2018. *Würzburg Irish Glosses*. online. PID https://wuerzburg.ie/.

Adrian Doyle. 2023. *Diplomatic Würzburg Glosses Treebank (DipWBG)*. Universal Dependencies 2.13. PID https://universaldependencies.org/treebanks/sga_dipwbg/.

Niall Ó Dónaill. 1977. *Foclóir Gaeilge-Béarla*. An Gúm. PID https://www.teanglann.ie/.

Kevin Scannell. 2018. *Droichead DIL*. Online. PID https://cadhan.com/droichead/. Presentation ”Is iomaí cor i saol an fhocail: Linking online dictionaries of Old Irish and Modern Irish”, NAACLT confrence, St. Louis, 2018.

David Stifter and Bernhard Bauer and Elliott Lash and Fangzhe Qiu and Nora White and Siobhán Barrett and Aaron Griffith and Romanas Bulatovas and Francesco Felici and Ellen Ganly and Truc Ha Nguyen and Lars Nooij. 2021. *Corpus PalaeoHibernicum*. Maynooth University, 1.0. PID http://chronhib.maynoothuniversity.ie.

Gregory Toner and Maxim Fomin and Grigory Bondarenko and Thomas Torma and Caoimhín Ó Dónaill and Hilary Lavelle. 2013-present. *An Electronic Dictionary of the Irish Language*. Royal Irish Academy. PID https://www.dil.ie. Based on the Contributions to a Dictionary of the Irish Language, 1913-1976.

Developing a Part-of-speech Tagger for Diplomatically Edited Old Irish Text

Adrian Doyle, John P. McCrae
Insight SFI Centre for Data Analytics, Data Science Institute, University of Galway
adrian.odubhghaill@universityofgalway.ie, john@mccr.ae

Abstract

POS-tagging is typically considered a fundamental text preprocessing task, with a variety of downstream NLP tasks and techniques being dependent on the availability of POS-tagged corpora. As such, POS-taggers are important precursors to further NLP tasks, and their accuracy can impact the potential accuracy of these dependent tasks. While a variety of POS-tagging methods have been developed which work well with modern languages, historical languages present orthographic and editorial challenges which require special attention. The effectiveness of POS-taggers developed for modern languages is reduced when applied to Old Irish, with its comparatively complex orthography and morphology. This paper examines some of the obstacles to POS-tagging Old Irish text, and shows that inconsistencies between extant annotated corpora reduce the quantity of data available for use in training POS-taggers. The development of a multi-layer neural network model for POS-tagging Old Irish text is described, and an experiment is detailed which demonstrates that this model outperforms a variety of off-the-shelf POS-taggers. Moreover, this model sets a new benchmark for POS-tagging diplomatically edited Old Irish text.

Keywords: Old Irish, POS-tagger, Multi-layer, Perceptron, Neural Network, Feature Engineering

1. Introduction

A part-of-speech (POS) tagger adds POS information to individual word and punctuation tokens which comprise a text. POS-taggers are generally employed early in the text preprocessing pipeline, typically being preceded only by tokenisation, though in some cases both tasks are carried out at the same time as a single initial step (Habash and Rambow, 2005). Many downstream NLP tasks, such as automatic term recognition (McCrae and Doyle, 2019) and coreference resolution (Darling et al., 2022), require text to be POS-tagged before they can be applied, and Yocum (2020, 89) claims that the lack of a POS-tagger for Old and Middle Irish has prevented the application of certain authorship attribution techniques to texts from the *Book of Leinster*. Therefore, POS-taggers are extremely important NLP tools which enable the application of a range of follow-on NLP techniques, and the lack of a POS-tagger for Old Irish is already hindering NLP research for the language.

Many types of POS-tagger have been developed over the decades, ranging from simple unigram taggers to complex deep learning models, and Schmid described a multi-layer perceptron (MLP) model for POS-tagging as early as 1994. Many taggers built more recently for a variety of languages still use comparable MLP approaches (Heigold et al., 2016; Hirpassa and Lehal, 2023; Mohammed, 2020; Tesfagergish and Kapočiūtė-Dzikienė, 2020). *The Natural Language Toolkit* (NLTK; Bird et al., 2009) includes several pre-built taggers as off-the-shelf solutions which need only to be trained on text data for a given language. This makes POS-tagging an achievable goal for any language for which training data is available. Generating a sufficient quantity of good quality text data to use for training such models can often be a significant obstacle to the creation of a POS-tagger (Chiche and Yitagesu, 2022, 18), however, particularly for under-resourced languages. This issue takes on another dimension in the case of historical languages like Old Irish, because no more text will ever be created by native speakers than whatever limited quantity has survived from the period in which the language was in use.

For Old Irish in particular several other factors also come into play. Tokenisation, for example, is a non-trivial task for Old Irish (Doyle et al., 2019). The primary reason for this is that words are not consistently separated by spacing in Old Irish. Instead, "... words which are grouped round a single chief stress and have a close syntactic connexion with each other are written as one in the manuscripts" (Thurneysen, 1946, 24). This makes the task of separating tokens difficult, and because tokenisation and POS-tagging are closely related tasks, this also leads to difficulty in POS-tagging.

A considerable amount of lexical variation in Old Irish texts also affects POS-tagging prospects. Old Irish manuscript orthography can be difficult to represent in modern digital editions (Doyle et al., 2018), and different editors represent various orthographic features in different ways. This leads to lexical variation in modern editions which is further increased by the typical spelling variation found in Old Irish manuscripts, and by the morphological complexity of the language. Heigold et al. note that “Morphologically rich languages exhibit large vocabulary sizes and relatively high out-of-vocabulary (OOV) rates on the word level” (2016, 1), which can cause problems for POS-taggers.

Little research to date has focused on POS-tagging for Old Irish, and no work has been published outlining a POS-tagger intended for use with diplomatically edited Old Irish text. The limited amount of work which has focused on Old Irish POS-tagging is discussed in section 3 of this paper. First, however, section 2 discusses the digital corpora which are available for Old Irish, outlining some of the difficulties these corpora create for prospects of developing a POS-tagger. Section 4 gives an overview of several off-the-shelf POS-taggers, before the development of a custom-built MLP model for POS-tagging diplomatically edited Old Irish text is described in section 5. An experiment to measure the accuracies of each of these models is outlined in section 6, and the results of this experiment are discussed in section 7.

Verb	**as·beir**	**do·beir**	**do·gní**

Ind.	*as·beir*
	as·biur
	as·ṁbeir
	as·ṁbiursa
	as·robair
		do·beir
		do·beirsem
		do·m-beir
			do·gní
			do·gni
			do·gníson

Dep.	*cenid·epersem*	*ceni·tabair* *ní·tabair* *·tabir*	*con·déni* *con·deni* *co·n-déni* *nád·ṅdéni* *ní·déni* *ní·deni* *ni·deni* *·ṅdéni* *·n-déni*

Table 1: Multiple dependent (**Dep.**) and independent (**Ind.**) forms of three Old Irish verbs (*as·beir*, *do·beir*, and *do·gní*) attested in the St. Gall glosses, all of which are analysed as `3sg.pres.ind.` by the St. Gall glosses database (Bauer et al., 2023).

2. Old Irish Text and Corpora

Old Irish refers to the historical stage of the Irish language as it was written from roughly the 7th to the 9th centuries. The majority of Old Irish text which survives in manuscripts dating from this Old Irish period is comprised of three collections of glosses; Würzburg (Wb.), Milan (Ml.), and St. Gall (Sg.). Between the three collections there are about 15,422 glosses written in Irish (Doyle, 2018; e-codices, 2005; Stifter et al., 2021), though these glosses are often very short with many being comprised of only a single word. Aside from these, a small amount of prose, poetry and miscellaneous glosses exist also. As such, the corpus of Old Irish which survives in contemporary sources is not particularly large by comparison to what is available for well resourced, modern languages. Adding to this, a number of factors compound to increase data sparsity within existing digital text repositories for Old Irish.

A considerable amount of code-switching between Old Irish and Latin occurs in each of the collections of glosses. Hence, any POS-tagger for Old Irish would likely be of limited utility if incapable of identifying and tagging Latin text to some extent as well as Old Irish. Spelling is inconsistent within the glosses, and a given word may be spelled multiple distinct ways, even by an individual scribe. The Latin content is variable also, and tends to show “the unusual orthographical peculiarities of Irish manuscripts” (Stokes and Strachan, 1901, xxiii).

Adding to this, Old Irish is morphologically very rich. The verbal complex in particular creates a considerable amount of lexical variability as verbs have both dependent and independent forms, and these forms can change radically in combination with various preverbs, conjunct and emphatic particles, and pronouns (see detailed discussion in McCone, 1997). Depending how verbs are tokenised, this variability can result in many distinct types of token, all representing the same grammatical expression of a single verb (see examples in table 1). Moreover, as an Insular Celtic language, a system of initial mutations can alter the anlaut of words in multiple grammatical situations, and this is expressed in the orthography. For example, the preposition *i* prefixes a nasal, *n*, to the word *degaid*, hence the combination *i ndegaid*. Therefore, both the beginnings and endings of words can change drastically in the orthography of Old Irish.

Further lexical variation is added into the mix by the regular use of abbreviations and contractions in Early Irish manuscripts. Some of these are used to represent set words, morphemes, and letters, such as the Tironian *et* (⁊), used to represent the conjunction *ocus* “and”, ł (Latin *vel*) to represent Irish *nó* “or”, and *·i·*, the Latin symbol representing *id est* (Irish *ed ón*). According to Thurneysen, other abbreviations can be “quite capricious” (1946, 25). Suspension strokes, for example, require a reader to determine from context the missing portion of an abbreviated word, and can therefore represent any number of potential character combinations. These abbreviations and contractions occur in manuscripts alongside the full forms of words in both Latin and Irish. To achieve a high degree of accuracy, therefore, a POS-tagger for Old Irish needs to be capable of tagging both the full forms of words as well as abbreviated and contracted forms.

The process of digitising Old Irish text invariably

Examples	Source	Gloss / Text	Raw Text	Tokens
1(a)	**SGP**	Sg. 1b1	“.i. ci insamlar”	“ci”, “in”, “in·samlar”
1(b)	**CorPH**	Sg. 1b1	“.i. ci in·samlar”	“.i.”, “ci’”, “in·”, “in·samlar”
2	**WBG**	Wb. 9a15	“.i. insamlatharside”	“.i.”, “in”, “samlathar”, “side”
3(a)	**SGP**	Sg. 194a1	“ocondṡruthsin”	“oco”, “nd”, “ṡruth”, “sin”
3(b)	**CorPH**	Sg. 194a1	“ocondṡruthsin”	“oco”, “ond”, “ṡruth”, “sin”
4	**MIDB**	Ml. 2b3	“.i. dintsruth”	“di”, “int”, “sruth”
5(a)	**SGP**	Sg. 7b8	“do·furgabtais”	“do”, “fur”, “-”, “do·furgabtais”
5(b)	**CorPH**	Sg. 7b8	“do·furgabtais”	“do·”, “·fur”, “∅”, “do·furgabtais”
6	**POMIC**	Arm. 64	–	“d-a-beir”, “side”, “0”
7	**WBG**	Wb. 24c16	“daberidsi”	“d”, “a”, “berid”, “si”
8(a)	**SGP**	Sg. 8a8	“da·ṅdichdet”	“d”, “a”, “ṅdi”, “ch”, “da·ṅdichdet”
8(b)	**CorPH**	Sg. 8a8	“da·ṅdichdet”	“d”, “a·”, “·ṅdi”, “ch”, “da·ṅdichdet”

Table 2: Examples of variation in tokenisation style between Early Irish text repositories: **SGP** (Bauer et al., 2023), **WBG** (Doyle, 2018), **MIDB** (Griffith, 2013), **POMIC** (Lash, 2014), **CorPH** (Stifter et al., 2021)

results in further lexical variation between the resulting corpora. Some modern editors aim to produce diplomatic editions, which resemble the text as it appears in the manuscript very closely. Such editors may make use of a large number of Unicode characters in order to represent manuscript features closely, which can result in a more sparse dataset. Other editors may attempt to correct manuscript errors, normalise spelling, supply missing text where manuscripts are damaged or deficient, expand abbreviations and contractions, and introduce ahistorical capitalisation and punctuation. The result is that the same text may be represented differently by two editions (see raw text for examples 1(a) and 1(b) in table 2).

Variation between Old Irish text repositories is even more apparent where tokenisation is applied. All three of the large corpora of glosses have been digitised and lexically annotated, and are available in online (Bauer et al., 2023; Doyle, 2018; Griffith, 2013; Stifter et al., 2021). Two Universal Dependencies (UD) treebanks exist, which contain a small number of POS-tagged and dependency parsed glosses from the Würzburg and St. Gall corpora (Doyle, 2023a,b), and the *Parsed Old and Middle Irish Corpus* (POMIC; Lash, 2014) contains a small amount of POS-tagged Old Irish prose. Each of these text repositories tokenise[1] Old Irish text in different ways, with the result that tokens from one repository are generally incompatible with those of another. Examples 3, 5 and 8 from table 2 demonstrate the same raw text being split into different tokens in accordance with the word-separation methods employed by different repositories[2]. Some repositories also include “empty” tokens representing parts-of-speech which are not realised in the orthography of the raw text (see examples 5(a), 5(b) and 6 in table 2). Finally, certain morphemes, as well as punctuation characters, are repeated in multiple tokens by some repositories, though they appear only once in the raw text (see examples 1, 3(b), 4, 5, and 8 in table 2).

As a result of these varied tokenisation methods, a POS-tagger trained on content from one repository could perform poorly even if tested on the same text content drawn from another repository, because the tokens encountered during training would not be the equivalents of those encountered during testing. This point is almost entirely moot, however, because, of all the repositories listed above, the only ones which share a single style of lexical annotation are the Würzburg glosses (Doyle, 2018) and the two small UD treebanks (Doyle, 2023a,b), all of which use UD-style POS-tags (Zeman, 2016). Aside from these, the only other repository which makes use of an established POS tag-set is POMIC (Lash, 2014), which utilises a variation of Penn-style POS-tags (Santorini, 1990) adapted originally for use with Old English (Santorini, 2016). All of the other text repositories (Bauer et al., 2023; Griffith, 2013; Stifter et al., 2021) use discrete lexical annotations. As such, POS data is not compatible between repositories, with the exception of the UD treebanks.

[1] The terms “token” and “tokenise” are used here in a general sense, referring to the division of text into word-like units which are thereafter annotated. Only Doyle (2018) actually utilises the terms “token” and “tokenisation”, however. Lash (2014) refers to “tokens” only once in POMIC’s annotation manual, but otherwise refers to “words” and “word-division” instead. As such, it would be unreasonable to expect the word divisions of most of these repositories to represent tokenisation in a traditional sense, or to expect tokens from one repository to match those of another.

[2] This point is made only to demonstrate that interoperability between resources is not easily possible. In the context of the methods utilised by individual repositories to divide text, each method is perfectly valid linguistically.

3. Related Work

Only a handful of attempts have been made to develop a POS-tagger for Early Irish. The earliest such attempt was made by Lynn (2012), who describes her model as a “fairly rudimentary” (2012, 23) prototype. Nevertheless, the production of this tagger was impressive as it predated the release of any corpus of lexically annotated Early Irish text. Lynn’s tagger was developed specifically for use with the text, *Táin Bó Fraích* (Meid, 1967), using a manually digitised version of the glossary which accompanied Meid’s print edition as a lexicon. Lynn describes how “The software reads previously unseen text, retrieves part-of-speech information from the machine-readable lexicon for each token in the text and subsequently inserts this information in the text as meta-data” (2012, 22). As the primary aim of Lynn’s work was to demonstrate the value of NLP tools for the field of Early Irish, no results detailing the accuracy of this POS-tagger were published. Presumably, as the lexicon was based on a glossary which had been specifically tailored to the vocabulary of the text used for testing, the tagger would struggle with OOV tokens if applied to unrestricted Old Irish text. Nevertheless, Lynn’s implementation demonstrated at an early stage that, with a sufficiently comprehensive machine-readable lexicon of attested word forms, a POS-tagger for Early Irish may be an achievable goal.

Bauer (2020) has claimed, during a seminar held by the Cardamom project group[3], to have achieved up to 75% accuracy when experimenting with off-the-shelf backoff taggers and Old Irish text drawn ultimately from *Corpus PalaeoHibernicum* (Stifter et al., 2021). Bauer was working with text from the *Annals of Ulster* and the St. Gall glosses. He achieved this 75% accuracy score working only with text from the *Annals of Ulster*, however, when text from St. Gall was included the highest overall accuracy achieved using a backoff tagger was about 30%. A higher overall accuracy of 54% was achieved using a Brill tagger (Brill, 1992). Bauer noted that tokens like preverbs were particularly problematic for tagging. Unfortunately, these results have not been published as of this writing[4].

The next attempt at creating a POS-tagger for Early Irish, and the first to be published in a decade, came when Darling et al. (2022) developed a tagger as a precursor to their work on coreference resolution for Old Irish. This tagger was trained and tested on text from POMIC (Lash, 2014). Normalisation was applied to the text to reduce orthographic variation (2022, 87). Further editing was carried out also, for example, new tokenisation had to be applied where Lash’s word-separation was unsuitable (2022, 87–88), and Lash’s POS-tags were simplified (2022, 88). Darling et al. utilised a Memory-Based Tagger, claiming “it is one of the most effective methods for developing a POS tagger from scratch, since it can learn from such specific features as initial and final characters as well as the context, yielding high rates of accuracy even for extremely small data sets” (2022, 88–89). Darling et al. carried out 10-fold cross-validation to evaluate the tagger, and report a global accuracy of 0.751 when accounting for both seen and unseen words (2022, 89). As texts in POMIC contain ahistorical punctuation, such as hyphenation within the verbal complex, and because Darling et al. had to apply further text normalisation, it is unclear how accurate this model might be if applied to diplomatically edited Old Irish text with more orthographic variation. With one in four words being tagged incorrectly, output from this tagger would still require considerable manual oversight to ensure quality. Nevertheless, these results are impressive given the relatively small amount of data available for training from POMIC. This work, therefore, represents a significant step towards the development of a generally useful tagger for Old Irish, particularly as this was the first such tagger to utilise an established POS tag-set like Penn (Santorini, 1990).

At the time of this writing, no other POS-taggers have been developed for use with Old Irish, and no further attempts have been made to improve POS-tagging prospects. No research has been published to date which addresses the prospect of tagging the type of text which might be found in more diplomatic editions, like *Thesaurus Palaeohibernicus* (Stokes and Strachan, 1901, 1903), and as diplomatic editions like these aim to closely represent Old Irish text as it appears in manuscript sources, this means that no tagger has yet been created which can POS-tag Old Irish as it was actually written. As a POS-tagger is a fundamental NLP tool, this leaves a considerable gap in the list of language resources which are currently available for Old Irish.

4. Baseline Methods

Several types of POS-tagger are available off-the-shelf, and each type may offer different benefits or drawbacks. This section gives an overview of each off-the-shelf model used in the experiment which will be detailed in section 6. As this experiment utilises text from UD treebanks, UDPipe’s bidirectional LSTM POS-tagger (Straka, 2018, 199) is a notable omission from the following list of models used. Unfortunately, no pre-trained UDPipe tagger

[3] https://cardamom-project.org/

[4] I would like to express my gratitude to Dr. Bauer for providing me with the relevant slides from his presentation, for discussing his results with me, and for permitting me to reference them here.

currently exists for Old Irish. Moreover, as UDPipe is an entire pipeline for processing `CoNLL-U` files, which includes other steps like tokenisation, a UDPipe tagger could not easily be tested in isolation as is required for this experiment. For these reasons it was not possible to include it in this experiment. The following models are all available through NLTK (Bird et al., 2009).

4.1. Unigram and N-gram Backoff Taggers

Functionally, NLTK's `UnigramTagger` model is the simplest used in this experiment. Bird et al. claim that "Unigram taggers are based on a simple statistical algorithm: for each token, assign the tag that is most likely for that particular token" (2009, 202). Unigram taggers learn specific tokens during training, and therefore, a weakness of these models is that they cannot assign a POS-tag to a token unless that specific token has been encountered during training. This is more problematic for languages like Old Irish, which have a high degree of lexical variation and hence higher OOV rates during testing. Because only the token which is being tagged is taken into consideration during tagging, another limitation of unigram taggers is that the context provided by surrounding words within a sentence is lost, and this can lead to poor results when tagging homographs (Bird et al., 2009, 203).

N-gram taggers, by contrast, can account for the context of a word within a sentence by looking at both the token and the POS-tags of the preceding *n* tokens. This functionality results in a data sparsity problem, however. N-gram taggers must see both a specific token and the preceding *n* POS-tags during training to be able to tag that same combination thereafter. "As *n* gets larger, the specificity of the contexts increases, as does the chance that the data we wish to tag contains contexts that were not present in the training data" (Bird et al., 2009, 205). An n-gram tagger may achieve higher accuracy than a unigram tagger for tokens which it has already seen in specific contexts, but there will be a larger number of tokens which it is incapable of tagging as a result of not having encountered them in particular contexts before. As with unigram taggers, this problem is exacerbated by languages like Old Irish with a high degree of lexical variation.

In order to alleviate the data sparsity issues caused by n-gram taggers, a common solution is to use them in combination with backoff taggers. If an n-gram tagger is unable to identify a POS-tag for a given token, having not seen it in a particular context during training, it will fall back on another POS-tagger model to tag the token instead. It is possible to use multiple layers of backoff taggers, and this is the approach which was used for the experiment detailed in this paper. Any time an n-gram tagger for which $n = x$ could not find a candidate POS-tag for a given token, the model would revert to another n-gram tagger for which the value of $n = x - 1$. This process of falling back on taggers with decreasing n-values would continue until the unigram tagger would finally reached. It was found that beginning with an n-value of $n = 3$ provided the best results.

4.2. Brill Tagger

The Brill tagger (Brill, 1992) is an inductive, transformation-based tagger. According to Bird et al. "Transformational joint classifiers work by creating an initial assignment of labels for the inputs, and then iteratively refining that assignment" (2009, 233). This improves upon n-gram taggers in a couple of ways. Firstly, Brill models can be much smaller than equivalent n-gram tagger models, as they do not need to store large, sparse arrays of n-grams. Secondly, as "The only information an n-gram tagger considers from prior context is tags, even though words themselves might be a useful source of information" (Bird et al., 2009, 208), a Brill tagger can take into account more contextual information. It can account for not only the tag of the preceding token, but also the token itself, and all the same information for the following token.

This functionality requires that the text must first be tagged by a more rudimentary POS-tagger. In the case of the implementation presented here, the unigram tagger described above was used for this purpose. As the Brill tagger trains on this pretagged text, instead of storing combinations of tag sequences which have occurred before, it instead develops a set of rules by which it alters certain tags depending on the preceding and following tokens.

4.3. Hidden Markov Model Tagger

Hidden Markov Model (HMM) taggers have comparable benefits to the Brill tagger in that they can take into account a wider range of token contexts than n-gram taggers. HMM taggers "assign scores to all of the possible sequences of part-of-speech tags" (Bird et al., 2009, 233), and then "choose the sequence whose overall score is highest". Like n-gram taggers, HMM taggers take into account both input tokens and the history of predicted tags. Unlike n-gram taggers, however, which use this kind of information to predict the best tag to apply to an individual token in a sequence, HMM taggers generate a probability distribution over tags, then calculate probability scores for sequences of tags by combining these probabilities. The sequence of tags with the highest probability score is chosen. In HMM taggers the HMM is applied in a discriminative manner, not as a generative model.

1.	The **token itself** (buffered, entirely lowercase)	6.	The **last five letters** of the token (all lowercase)
2.	Whether the **token is entirely lowercase** in the sentence (Boolean: true/false)	7.	Whether the **token occurred first** in the sentence (Boolean: true/false)
3.	Whether the **token is entirely capitalised** in the sentence (Boolean: true/false)	8.	Whether the **token occurred last** in the sentence (Boolean: true/false)
4.	Whether the **first letter of the token is capitalised** in the sentence (Boolean: true/false)	9.	The **previous two tokens** (entirely lowercase)
5.	The **first five letters** of the token (all lowercase)	10.	The **following two tokens** (entirely lowercase)

Table 3: Features Collected for Each Token as Input for the MLP Tagger.

4.4. Perceptron Tagger

The perceptron tagger used in this experiment was first implemented by Honnibal and ported over to NLTK from *TextBlob* (2013). It is a neural model which takes various inputs, called features, and uses these to predict the best POS candidate for a given token. According to Honnibal, these features 'will be things like "part of speech at word i-1", "last three letters of word at i+1" etc'. As the model is trained to associate particular features it receives as input with parts-of-speech the weights connecting the various inputs and outputs within the model are increased and decreased in accordance with how useful the model determines they are in aiding it to complete its task. The power of the perceptron tagger to exploit the context of surrounding tokens comes from the features used as input, and the model's own ability to regulate the importance of each of these features as it trains.

5. Methodology

In their review of state-of-the-art POS-tagging solutions, Chiche and Yitagesu concluded that "the use of deep learning (DL) oriented methodologies improves the efficiency and effectiveness of POS tagging in terms of accuracy and reduction in false-positive rate" (2022, 21–22). Several recent papers corroborate this finding, and demonstrate that MLP models often perform well in under-resourced and morphologically rich language settings (Heigold et al., 2016; Hirpassa and Lehal, 2023; Mohammed, 2020; Tesfagergish and Kapočiūtė-Dzikienė, 2020). For this reason a custom MLP tagger was developed for this experiment.

This model differs from the perceptron tagger in a couple of key ways. Firstly, the hidden layers of the MLP tagger should enable it to adapt to non-linearly separable data extracted from the Old Irish text. Secondly, feature engineering for the MLP tagger was customised to focus the attention of the model on aspects of the text which were expected to provide better POS-tagging performance specifically for Old Irish morphology. These aspects were then assessed during ablation analysis to ensure that they did, in fact, provide benefits. For the purpose of feature engineering, ten features were collected from the text for each token (see table 3).

The first feature collected is the token itself. This token is rendered in lowercase to reduce lexical variation, and is then buffered to ensure that all tokens will be of the same length. As the token is rendered entirely in lowercase, features 2 to 4 in table 3 provide information to the model regarding letter case as it is used in the text. Capitalisation does not mark particular parts-of-speech in Old Irish manuscripts, nor hence in diplomatic editions, as it does in modern orthographies, for example, with proper nouns in English or all nouns in German. Capital letters are occasionally employed, however, to match rare manuscript usage of majuscule letters. Majuscule letters are typically employed in manuscripts from this period only at the beginning of paragraphs or significant sections of text, though more than one majuscule letter may be used in sequence. An example of this, drawn from the St. Gall manuscript, can be seen in figure 1, where the initial word of a poem is written entirely using majuscule letters. Given this atypical usage of capitalisation by comparison to modern European orthographies, it was unclear what effect would be produced by either the inclusion or exclusion of features 2, 3 and 4 until ablation analysis was conducted, however, as "POS tagging literature has tonnes of intricate features sensitive to case" (Honnibal, 2013), they were included for this experiment. Their inclusion may also make this POS-tagger more flexible, and better capable of handling less diplomatically edited Old Irish text, where editors employ capitalisation in accordance with modern standards.

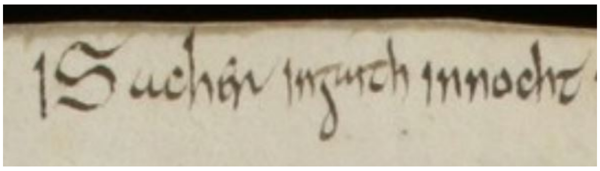


Figure 1: *IS acher ingáith innocht* - St. Gallen, Stiftsbibliothek, Cod. Sang. 904, f. 194 (www.e-codices.ch).

As has been discussed in section 2, both the beginnings and endings of Old Irish words can change drastically in certain grammatical situations. For

	MLP	MLPlus
Hidden layers	3	3
Neurons Per Hidden Layer	64	64
Hidden Layer Activation	ReLU	ReLU
Dropout	20%	20%
Output Layer Activation	softmax	softmax
Training Epochs	50	50
Early Stopping Patience (Epochs)	7	7
Optimiser	Adam	Adam
Rule-based Reassignment Layer	No	Yes

Table 4: Parameters for MLP and MLPlus Taggers.

this reason features 5 and 6 in table 3 focus the attention of the model on the first and last five letters, respectively, of each token. While feature 6 captures morphological information common to many languages, such as case endings for nouns and subject inflections for verbs, feature 5 is intended to cater more specifically to aspects of Old Irish morphology, like initial mutations. Strictly speaking, features 5 and 6 are not individual features themselves, but are comprised of 5 sub-features each. For each token, not only are the first and last five letters collected in combination, but also the first and last four, three, and two letters in combination, as well as the initial and final letters on their own. Therefore, for the word *disruthaigedar*, the following ten sub-features would be collected: *d*, *di*, *dis*, *disr*, *disru*, *r*, *ar*, *dar*, *edar*, and *gedar*. Next, each of these ten sub-features are rendered in lowercase and buffered, like tokens collected for feature 1.

Features 7 to 10 in table 3 relate to the placement of a given token both within the sentence, and relative to other tokens. This kind of information can be helpful in determining POS-tags, as some parts-of-speech are more likely to occur in combination with certain other parts-of-speech. Determiners and adjectives, for example, often occur in combination with nouns, while preverbs and conjunct particles typically precede verbs. Features such as these are not uncommon in POS-taggers, and are also used by Honnibal for his tagger (2013).

Once collected for each token, all ten features were vectorised and one-hot encoded using the `DictVectorizer` class from the `sklearn.feature_extraction` module (Pedregosa et al., 2011). At this point they could be used as input for the model.

Experimentation with hyperparameters during training revealed that the best results were achieved using three hidden layers, with sixty-four neurons per hidden layer. For hidden layers, `ReLU` was used as the activation function, and the `softmax` function was used in the output layer. Optimisation was performed using the `Adam` method (Kingma and Ba, 2015). To avoid overfitting during training, a dropout rate of 20% was used on all hidden layers and early stopping was applied. Validation loss was tracked as a metric to determine when early stopping should occur, and model weights were returned to those which achieved the minimum validation loss during training. An overview of model parameters can be found in table 4.

As the results in section 7 will show, this MLP model performed well relative to other taggers, however, for certain POS-tags which occur particularly infrequently within the corpus, its performance suffered. For this reason the MLPlus model was created. This tagger is almost identical to the first MLP model, except that a rule-based layer is added at the end of the tagging pipeline which reassigns POS-tags for certain tokens. During model training, tokens from the training set which are labeled as interjections, proper nouns, or punctuation are collected. Those which are not homonymous with other tokens which represent more common parts-of-speech are stored in an `infrequent POS-tags` list. When the MLPlus tagger is used during testing it first predicts POS-tags for all tokens, as the MLP model would. Next, a script compares every token in the model's output against each token in the `infrequent POS-tags` list. If a token from the output matches a token in the `infrequent POS-tags` list, the predicted POS-tag is replaced with the POS-tag from the list.

6. The Experiment

6.1. The Data

As has been discussed in section 2, tokenisation methods vary between lexically annotated Old Irish text repositories, and few repositories utilise common POS tag-sets like Penn (Santorini, 1990) and UD (Zeman, 2016). This limits the text available for use in this experiment to either the Old Irish content of POMIC (Lash, 2014), or that of the UD treebanks (Doyle, 2023a,b). Because the text of both of the UD treebanks is diplomatically edited, it was preferable to use UD content in this experiment. It was not possible to also include annotated content from POMIC because this resource separates words differently to the UD treebanks, and utilises a different POS tag-set. This limits the scope of this experiment to diplomatically edited gloss content and a small quantity of poetry. Though it would be preferable to incorporate other genres of text in this experiment, and perhaps text edited to different standards also, the lack of any other corpus which has been tokenised and annotated so as to be compatible with the UD treebanks has ruled out this possibility for now.

The UD corpora are both quite small, with a combined extent of only ninety-eight glosses at the time of this writing. This would not be sufficient to train a POS-tagger, particularly an MLP model. Fortunately, while the master branch of the St. Gall treebank contains only sixty-four glosses at present, the remainder of the corpus has been POS-tagged and annotated with morphological features. This data is stored in the `incomplete.conllu` file which can be found in the development branch[5] of the treebank. Taking into account this content, there are 3,469 POS-tagged glosses containing 21,749 tokens. This should be sufficient to train a reasonably accurate POS-tagger, even on diplomatically edited text. Moreover, Latin tokens in these glosses are are all POS-tagged `X` and annotated with the morphological feature `Foreign=Yes`. This should give taggers the opportunity to learn to distinguish between Latin and Irish text.

6.2. Testing the Models

Because the contents of the St. Gall glosses tend to reflect the thematic context of the Priscian chapter to which they relate, k-fold cross-validation could result in a high number of OOV words unless all glosses within the corpus were shuffled randomly. Instead of randomising all of the data and passing over it sequentially, this experiment uses Monte Carlo cross-validation in order to get a clear picture of each tagger's ability to cope with unseen Old Irish text. This approach required carrying out several passes over the dataset, with each POS-tagger being trained on the same data each pass, then tested on the same test set also.

1,000 passes were carried out in total to ensure the accuracy of the results, while limiting the computational expense of the experiment to a tolerable level. For each pass, 5% of all glosses were split off at random to be used as a test set, and the remainder would serve as the training set. For the MLP and MLPlus taggers, a further 10% of glosses were split from the remainder of the training set at random each pass to be used as a validation set. After all passes for a tagger were complete, the accuracy scores for all passes were averaged to generate the tagger's overall average POS-tagging accuracy. The average accuracy of each tagger over 1,000 passes for each POS-tag can be found in table 5, as well as the total average accuracy for all tokens.

7. Results

As can be seen in table 5, the the unigram and n-gram taggers achieved the lowest scores, 0.698 and 0.708 respectively. The Brill tagger scored marginally better than these, with an accuracy of 0.726. The HMM tagger showed a reasonable improvement over the first three models, with an overall accuracy score of 0.783, and it achieved the highest accuracy scores of any model for tagging determiners and particles specifically. This may speak to the value of calculating probabilities for POS distributions for languages with a lot of lexical variation, over approaches which either rely or fall back on using lookup tables for specific tokens.

The three neural network models offer considerable improvements over all of the other taggers. NLTK's perceptron tagger boasts an 8.5% improvement over the next best performing model, and the MLP model improves upon that by another 2.8%. As has been noted above, the MLP tagger seems to have suffered from under-representation of three particular POS-tags in the data used for this experiment. Only seven tokens were tagged `PUNCT`[6], eighteen were tagged `INTJ` and fifty-four were tagged `PROPN`. The rule-based reassignment layer of the MLPlus tagger seems to have alleviated this issue somewhat as this model achieved the highest accuracy score for `PUNCT`, and showed a marginal improvement for `PROPN`. As these POS-tags represent such a small percentage of the dataset, however, these POS-level improvements do not translate to a significant increase in overall accuracy for the MLPlus tagger. No improvement in overall accuracy can be seen in table 5 as results there are limited to three decimal places. Nevertheless, the MLPlus model is the best performing tagger in most POS categories.

7.1. Ablation Analysis

Ablation analysis carried out on the MLP tagger determined that most of the features outlined in table 3 are beneficial for POS-tagging diplomatically edited Old Irish text, and none hinder the model's performance. It was found that accuracy drops significantly to 0.768 if only the buffered, lowercase token is used as input. Conversely, accuracy remains at 0.896 when features pertaining to letter case (2, 3 and 4 in table 3) are removed from the feature-set. This is to be expected as capitalisation does not mark particular parts-of-speech in Old Irish manuscripts (see discussion in section 5).

Accuracy drops to 0.826 if the feature-set does not include the first and last five letters of each token (features 5 and 6 in table 3), which indicates the value of this morphological information for POS-tagging. Though Honnibal used only the last three letters of tokens as features for his POS-tagger (2013), it was found during experimentation that that capturing up to five letters at the beginning and end of each word produced the best results for the Old Irish text used in this experiment. Using fewer resulted in accuracy drops between 2% and

[5] `https://github.com/UniversalDependencies/UD_Old_Irish-DipSGG/tree/dev/not-to-release`

[6] More punctuation has been included in the latest version of the St. Gall glosses treebank (Doyle, 2023a).

	Unigram	N-gram: n=3	Brill	HMM	Perceptron	MLP	MLPlus
ADJ	0.526	0.530	0.527	0.575	0.694	**0.862**	**0.862**
ADP	0.867	0.825	0.876	0.855	0.893	**0.927**	**0.927**
ADV	0.982	0.982	0.981	0.975	0.974	**0.990**	**0.990**
AUX	0.815	0.831	0.847	0.873	**0.910**	0.896	0.896
CCONJ	0.971	0.966	0.950	0.834	0.956	**0.999**	**0.999**
DET	0.789	0.880	0.886	**0.928**	0.922	0.918	0.918
INTJ	0.656	0.666	**0.678**	0.522	**0.678**	0.000	0.000
NOUN	0.610	0.619	0.612	0.675	0.899	**0.906**	**0.906**
NUM	0.764	**0.790**	0.779	0.703	0.724	0.718	0.718
PART	0.615	0.667	0.775	**0.840**	0.833	0.814	0.814
PRON	0.791	0.747	0.817	0.628	0.814	**0.909**	**0.909**
PROPN	0.121	0.118	**0.124**	0.001	0.055	0.000	0.001
PUNCT	**1.000**	**1.000**	**1.000**	0.415	**1.000**	0.000	**1.000**
SCONJ	0.746	0.790	0.837	0.848	**0.861**	0.832	0.832
VERB	0.532	0.525	0.524	0.776	0.814	**0.880**	**0.880**
X	0.542	0.563	0.566	0.765	0.846	**0.886**	**0.886**
Total Average	0.698	0.708	0.726	0.783	0.868	**0.896**	**0.896**

Table 5: Average POS-tagging Accuracy for all Taggers after 1,000 Training Passes. Best Result per Category in **Bold** and **Underlined**.

7%. This seems to indicate that morphologically significant information for POS-tagging Old Irish penetrates deeper into tokens than is typical of other languages. This can be seen, for example, in the endings of deponent verbs like *suidigidir*, *foilsigidir*, and *cruthaigidir*.

Removing features which inform the model whether a token occurred first or last in a sentence (7 and 8 in table 3) does not appear to affect performance, as the accuracy remains at 0.896. Removing information regarding the following and preceding tokens (features 9 and 10 in table 3), however, drops the accuracy to 0.845.

8. Future Work

Future avenues of research may seek to achieve higher tagging accuracy than the MLP and MLPlus models outlined in this paper by utilising them in combination with other models which require text to be pre-tagged, like the Brill tagger. Though it performed well when tagging punctuation for the dataset used in this experiment, the MLPlus model may be bolstered by supplementing the `infrequent POS-tags` list with a combination of common punctuation characters, and approximations of common manuscript punctuation (such as :⌐, ·~, and .,.,.,) and other symbols (see Groenewegen, 2011). Finally, it is possible that another variety of MLP approach may prove more successful on Old Irish data. Though Heigold et al. found that, for morphologically rich languages, “As long as carefully tuned neural networks of sufficient capacity (e.g., number of hidden layers) are used, the effect of the specific network architecture (e.g., convolutional vs. recurrent) is small for the task under consideration” (2016), more recently Tesfagergish and Kapočiūtė-Dzikienė (2020) have found that a bidirectional LSTM tagger showed notably improved accuracy for Northern-Ethiopic Languages, and Hirpassa and Lehal (2023) found that a variety of bidirectional LSTM tagger performed best for the Amharic Language. It is therefore possible that improvements might be sought over the MLP models presented here by developing a bidirectional LSTM tagger for Old Irish.

9. Conclusion

This paper has described the training of five off-the-shelf POS-taggers, as well as the development and training of two custom-built MLP taggers, on a corpus of diplomatically edited Old Irish text. A comparison of tagging accuracies achieved by these taggers shows that the custom-built MLPlus tagger is the best performing overall, as well as in nine out of sixteen individual POS categories.

A direct comparison cannot be drawn between the scores achieved by taggers used in this experiment and the global accuracy of 0.751 reported by Darling et al. (2022, 89), as each of these experiments utilised not only different corpora of text, but an entirely different POS tag-set. Given the nature of the text data used for this experiment, however, it seems reasonable to suggest that the MLPlus model has set the first benchmark for POS-tagging diplomatically edited Old Irish text.

10. Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Clodagh Downey, whose expertise has been invaluable during the course of this research. Without her guidance and keen attention to detail the current work would not be what it is. Any remaining errors and omissions are entirely my own.

This work has been possible thanks to the support of the Science Foundation Ireland (SFI) as part of Grant Number SFI/12/RC/2289_P2 Insight_2, Insight SFI Centre for Data Analytics. It has also been funded by the University of Galway through the Digital Arts and Humanities Programme, and by the Irish Research Council through the Government of Ireland Postgraduate Scholarship Programme.

11. Bibliographical References

Bernhard Bauer. 2020. ChronHib, CorPH and the Corphusator: Building an Early Irish Corpus. Unpublished.

Bernhard Bauer, Rijcklof Hofman, and Pádraic Moran. 2023. St Gall Priscian Glosses, version 2.1. Accessed: February 12, 2024.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly, Sebastopol.

Eric Brill. 1992. A Simple Rule-Based Part of Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, page 152–155, USA. Association for Computational Linguistics.

Alebachew Chiche and Betselot Yitagesu. 2022. Part of Speech Tagging: a Systematic Review of Deep Learning and Machine Learning Approaches. *Journal of Big Data*, 9.

Mark Darling, Marieke Meelen, and David Willis. 2022. Towards Coreference Resolution for Early Irish. In *Proceedings of the 4th Celtic Language Technology Workshop within LREC2022*, pages 85–93, Marseille, France. European Language Resources Association.

Adrian Doyle. 2018. Würzburg Irish Glosses. Accessed: February 12, 2024.

Adrian Doyle, John P. McCrae, and Clodagh Downey. 2018. Preservation of Original Orthography in the Construction of an Old Irish Corpus. In *Proceedings of the LREC 2018 Workshop: "CCURL2018 – Sustaining Knowledge Diversity in the Digital Age"*, pages 67–70, Miyazaki, Japan.

Adrian Doyle, John P. McCrae, and Clodagh Downey. 2019. A Character-Level LSTM Network Model for Tokenizing the Old Irish text of the Würzburg Glosses on the Pauline Epistles. In *Proceedings of the Celtic Language Technology Workshop*, pages 70–79, Dublin, Ireland. European Association for Machine Translation.

e-codices. 2005. e-codices - Virtual Manuscript Library of Switzerland. Accessed: February 12, 2024.

Aaron Griffith. 2013. A Dictionary of the Old-Irish Glosses. Accessed: February 12, 2024.

Dennis Groenewegen. 2011. Tionscadal na Nod. Accessed: February 21, 2024.

Nizar Habash and Owen Rambow. 2005. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 573–580, Ann Arbor, Michigan. Association for Computational Linguistics.

Georg Heigold, Günter Neumann, and Josef van Genabith. 2016. Neural Morphological Tagging from Characters for Morphologically Rich Languages. *ArXiv*, abs/1606.06640.

Sintayehu Hirpassa and G.S. Lehal. 2023. Improving part-of-speech Tagging in Amharic Language Using Deep Neural Network. *Heliyon*, 9(7):e17175.

Matthew Honnibal. 2013. A Good Part-of-Speech Tagger in about 200 Lines of Python. Accessed: February 21, 2024.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Elliott Lash. 2014. POMIC Annotation Manual. Manual, The Dublin Institute for Advanced Studies.

Teresa Lynn. 2012. Medieval Irish and Computational Linguistics. *Australian Celtic Journal*, 10:13–27.

Kim McCone. 1997. *The Early Irish Verb*, 2 edition. An Sagart, Maynooth.

John P. McCrae and Adrian Doyle. 2019. Adapting Term Recognition to an Under-Resourced Language: the Case of Irish. In *Proceedings of the Celtic Language Technology Workshop*, pages 48–57, "Dublin, Ireland. European Association for Machine Translation.

Wolfgang Meid, editor. 1967. *Táin Bó Fraích*. The Dublin Institute for Advanced Studies, Dublin.

Siraj Mohammed. 2020. Using Machine Learning to Build POS tagger for Under-resourced Language: The Case of Somali. *International Journal of Information Technology*, 12.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Beatrice Santorini. 1990. Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision). Standard, Department of Computer and Information Science, University of Pennsylvania.

Beatrice Santorini. 2016. Annotation Manual for the Penn Historical Corpora and the York-Helsinki Corpus of Early English Correspondence. Accessed: February 19, 2024.

Helmut Schmid. 1994. Part-of-Speech Tagging With Neural Networks. In *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*, Kyoto, Japan.

David Stifter, Bernhard Bauer, Elliott Lash, Fangzhe Qiu, Nora White, Siobhán Barrett, Aaron Griffith, Romanas Bulatovas, Ellen Felici, Francesco abd Ganly, Truc Ha Nguyen, and Lars Nooij. 2021. Corpus PalaeoHibernicum (CorPH) v1.0. Accessed: February 12, 2024.

Whitley Stokes and John Strachan, editors. 1901. *Thesaurus Palaeohibernicus*, volume 1. The Dublin Institute for Advanced Studies, Dublin.

Whitley Stokes and John Strachan, editors. 1903. *Thesaurus Palaeohibernicus*, 2 edition, volume 2. The Dublin Institute for Advanced Studies, Dublin.

Milan Straka. 2018. UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Senait Gebremichael Tesfagergish and Jurgita Kapočiūtė-Dzikienė. 2020. Part-of-Speech Tagging via Deep Neural Networks for Northern-Ethiopic Languages. *Information Technology and Control*, 49(4):482–494.

Rudolf Thurneysen. 1946. *A Grammar of Old Irish*, 2 edition. The Dublin Institute for Advanced Studies, Dublin.

Christopher Guy Yocum. 2020. Text Clustering and Methods in the Book of Leinster. In Elliott Lash, Fangzhe Qiu, and David Stifter, editors, *Morphosyntactic Variation in Medieval Celtic Languages. Corpus-Based Approaches*, pages 85–111. De Gruyter Mouton, Berlin.

Dan Zeman. 2016. UD Guidelines V2. Accessed: February 19, 2024.

12. Language Resource References

Doyle, Adrian. 2023a. *Diplomatic St. Gall Glosses Treebank*. Universal Dependencies. Accessed: February 19, 2024.

Doyle, Adrian. 2023b. *Diplomatic Würzburg Glosses Treebank*. Universal Dependencies. Accessed: February 19, 2024.

Lash, Elliott. 2014. *The Parsed Old and Middle Irish Corpus (POMIC). Version 0.1*. The Dublin Institute for Advanced Studies. Accessed: February 12, 2024.

From YCOE to UD: rule-based root identification in Old English

Luca Brigada Villa, Martina Giarda
University of Pavia/Bergamo, University of Pavia/Bergamo
{luca.brigadavilla, martina.giarda}@unibg.it

Abstract

In this paper we apply a set of rules to identify the root of a dependency tree, following the Universal Dependencies formalism and starting from the constituency annotation of the York-Toronto-Helsinki Parsed Corpus of Old English Prose (YCOE). This rule-based root-identification task represents the first step towards a rule-based automatic conversion of this valuable resource into the UD format. After presenting Old English and the annotated resources available for this language, we describe the different rules we applied and then we discuss the results and the errors.

Keywords: Old English, root-identification, YCOE, Universal Dependencies

1. Introduction

The York-Toronto-Helsinki Parsed Corpus of Old English Prose (henceforth YCOE) (Taylor et al., 2003) is the reference treebank for studies on Old English syntax. It is a 1.5-million-word constituency treebank, annotated following the Penn format. As a sister corpus to the Penn-Helsinki Parsed Corpus of Middle English (PPCME2) (Kroch and Taylor, 2000), it uses the same form of annotation and is accessed by the same search engine, *CorpusSearch2*, whose usage is not always intuitive. Moreover, dependency annotation schemes have gained widespread acceptance, making the Universal Dependencies (UD) format, as described in de Marneffe et al. (2021), the standard for dependency treebanks. In the latest released version (May 15, 2023), more than 245 treebanks in 141 languages (both modern and ancient) were annotated according to UD standards. However, no treebank for Old English is available in dependency format, in contrast to the large amount of annotated resources for Present-Day English.[1]

These considerations led us to attempt the creation of a dependency treebank of Old English, following the UD format. Training a monolingual parser would require a large sample of manually annotated data, which can be really time-consuming to produce. After attempting to train a multilingual parser (Brigada Villa and Giarda, 2023), we aimed to produce a rule-based conversion of the YCOE, so that the massive work of the creators of this treebank would not have been lost. The starting point of this conversion is a rule-based root identification task, since the root is the node from which every other depends. Using the original Part-of-Speech (POS) tags in the YCOE, we created hierarchical rules to identify the root of the sentences. Afterwards, we checked the efficacy of these rules against a manually annotated gold set.

The paper is structured as follows: in Section 2 we introduce Old English providing a brief description of its history, developments, and morpho-syntactic features. Moreover, we provide a brief overlook of the main available resources for this language and a description of the YCOE structure. In Section 3 we present our data and methodology, whereas Section 4 is dedicated to the results and Section 5 to error discussion. Finally, Section 6 concludes the paper and summarizes our findings.

2. Old English

Old English is a West-Germanic language, classified with Old Frisian and Old Saxon among the so-called Ingvaeonic languages. It was the language spoken in England after Angles, Saxons, Jutes and Frisians came to Britain and settled in the island in the 5th century. It is attested from the 7th century, except for some older brief runic inscriptions, whereas its ending point is conventionally established in 1066, date of the Norman Conquest of England (von Mengden, 2017a). Old English is a fusional language with inflectional word classes. As other Germanic languages, it has two main conjugational system, called, respectively, strong and weak verbs. Strong verbs build the preterit by means of apophony, i.e. vowel alternation, also found in Present-Day English (PDE) irregular verbs, whereas weak verbs insert a dental suffix, just as PDE regular verb, whose past form is constructed with the *-ed* suffix. Finite Old English verbs inflect for mood (indicative, subjunctive, imperative), tense (present and past), number, and person. All the plural forms in all moods and tenses, and the first and third person singular in the subjunctive show syncretism (von Mengden, 2017b). Concerning word order, it is not as rigid as in PDE, despite the fact that some regularities can be found (Mitchell and Robinson, 2012: 63-65). It is still debated whether the basic word order was (S)VO or (S)OV.

[1] UD has 10 different treebanks for Present-Day English.

(Molencki, 2017: 101): it is generally assumed the early stages of the Old English language were characterised by a competition between (S)OV and (S)VO word orders, in which the former prevailed over the latter as the basic order. (Fischer et al., 2001: 51; Pintzuk and Taylor, 2006). Like other ancient and modern Germanic languages, OE also exhibits V2, i.e. the tendency of the finite verb to follow the first constituent, regardless of its type. Nouns are inflected by number and case, following three inflectional classes, depending on their original Proto-Germanic stem. Old English retains four of the eight original Indo-European cases: nominative, accusative, genitive, and dative. Moreover, residual traces of the instrumental are found. Depending on the class, different cases can show syncretism. Concerning the order of other constituents in the NP, nouns are generally preceded by modifiers, e.g. demonstratives, adjectives, genitive complements. However the latter can follow the noun if another preceding modifier is present. In PPs, adpositions tend to precede a noun, but generally follow a pronoun; however, the opposite is also attested (Molencki, 2017).

Old English allows subjectless constructions, above all with reference to natural phenomena. However, it has also developed the use of empty pronominal subjects (*hit* 'it' and *þær* 'there'), which were neither anaphoric or cataphoric (Molencki, 2017: 104). Old English exhibits some complex (or periphrastic) verbal constructions, whose origin and grammaticalization are still debated among scholars. Both present and past perfect were made of the auxiliary *habban* 'have' (for transitives) or *beon/wesan* (for intransitives) and the past participle of the main verb, this latter either inflected or not (Molencki, 2017: 112-113). The passive voice was also expressed by a periphrastic construction, made of the auxiliary *beon/wesan* 'be' or *weorþan* 'become' and the past participle, with the sole exception of the verb *hatan* 'be called' (but also 'order'). A part for asyndetic clauses, Old English texts are richer in paratactic devices (very often repetitive) than in subordination. However, the borderline between parataxis and hypotaxis is rather vague, above all in temporal clauses, in which the sequence of events is often expressed by means of clause-initial *þa* 'then'. (Molencki, 2017: 117).

2.1. Annotated resources for Old English

Differently from other ancient languages, such as Latin or Ancient Greek,[2] and its contemporary counterpart, scholars have devoted little attention to the creation of resources to study Old English. At the moment, the sole syntactically annotated resources for this language are the constituency treebank YCOE and its poetry counterpart, the York-Helsinki Parsed Corpus of Old English Poetry (YCOEP) (Pintzuk and Plug, 2002), which follow the Penn style. Despite their value in size, these treebanks are hardly machine- nor user-friendly, have no interface and can only be investigated through their search engine *CorpusSearch2*, which requires an intensive training in order to write even simple queries.

A first attempt to build a UD treebank for Old English has been made by Arista (2022a) and Arista (2022b), but the treebank has not been published yet. Also, Brigada Villa and Giarda (2023), trained multilingual parser on data from Old English, Modern German, Modern Icelandic and Modern Swedish data to parse Old English. However, no attempts at a rule-based conversion of the whole YCOE have been made yet. There exists a pipeline to convert Penn-format constituency treebanks into UD dependency treebanks (Arnardóttir et al., 2020): however this is designed for the Icelandic Parsed Historical Corpus (IcePaHC; Rögnvaldsson et al., 2012) and the Faroese Parsed Historical Corpus (FarPaHC; Ingason et al., 2014), which, though based on the Penn Parsed Corpora of Historical English (PPCHE, also base for the YCOE; Kroch and Taylor, 2000), present some crucial differences in the annotation scheme, some of which would require a more thoroughful revision.

2.1.1. YCOE description

The YCOE is a 1.5 million word syntactically-annotated corpus. Its size and representativeness makes it a valuable resource for the study of Old English. However, the constituency format and the lack of some information (e.g. lemmatization, and some morphological features) may hinder data retrieval. A conversion of this treebank into the Universal Dependencies format would solve some of the problems, while preserving the huge amount of data already available. The format in which the sentences in the YCOE treebank are parsed consists of a limited hierarchical bracketing comprising labeled parentheses to represent syntactic trees. Word forms serve as the fundamental units of the sentence: they are POS tagged and then grouped together to construct more complex structures such as phrases and sentences. Each element within the sentence is labeled, enabling the retrieval of the tree structure from the annotation.[3]

An example of annotation can be found in Figure 1. In this sentence, we can notice that the words are

[2] The latest release of UD (v2.11) includes 5 treebanks for Latin and 2 for Ancient Greek.

[3] all POS tags are retrievable here: https://www-users.york.ac.uk/~lang22/YCOE/doc/annotation/YcoeLite.htm#pos_labels, whereas the syntactic tags can be found here: https://www-users.york.ac.uk/~lang22/YCOE/doc/annotation/YcoeLite.htm#syntactic_labels.

```
( (IP-MAT (NP-NOM (PRO^N hi))
          (VBDI oncneowon)
          (ADVP-TMP (ADV^T +da))
          (CP-THT (C +d+at)
                  (IP-SUB (NP-NOM (PRO^N hi))
                          (ADJP-NOM-PRD (ADJ^N nacode))
                          (BEDI w+aron)))
          (. ,))
  (ID cootest,Gen:3.7.132))
```

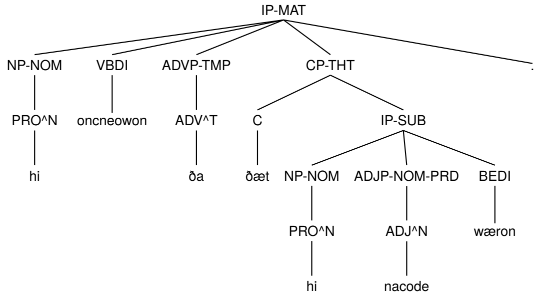


Figure 1: YCOE annotation style of the sentence `cootest,Gen:3.7.132`, whose translation is ‘Then they realized that they were naked’

the innermost elements in the hierarchical structure (*hi*, *oncneowon*, *+da*, *+d+at*, *hi*, *nacode* and *w+aron*) and phrases can contain either words or smaller phrases (`NP-NOM`, `ADVP-TMP`, `CP-THT`, `IP-SUB`, `NP-NOM`, `ADJP-NOM-PRD`). Wrapping all the words and phrases of the sentence, there is a label denoting a clause (in this case `IP-MAT`).

3. Data and Methodology

Our data[4] consists of 390 manually annotated sentences, from three different texts: *Adrian and Ritheus*, the first homily of Ælfric’s *Supplemental Homilies*, and the first 100 sentences of Book 1 of Bede’s *Historia Ecclesiastica Gentis Anglorum*, translated in Old English.[5] The choice to include also Bede is due to the fact that Latin has had a great influence on Old English syntax, pushing it toward a more frequent use of hypotaxis. Since a significant part of the Old English corpus is made of translations, we wanted to test our rule-base conversion both on translations and texts originally written in Old English, without a Latin source.

The set of sentences selected to conduct this study was manually annotated following the Universal Dependencies guidelines. This set formed our gold standard and was used to compare the annotations performed by our model.

The YCOE has the tendency to split coordinated clauses into different sentences, with different sentence IDs. According to context, some coordinated sentences have been connected to their main clause. Although punctuation is not always reliable, since it is added by the modern editor, the general rule was to connect clauses divided by commas, but to leave separated those divided by a period, even if the following sentence started with the conjunctions *and* ‘and’ or *ac* ‘but’. The sentences divided by a semicolon have been treated differently depending on the context. In all cases, the sentence ID of the main clause was retained.

In this section, we will discuss the two main steps of our process: (1) the conversion from the format in which the YCOE treebank appeared into the CoNLL-U format (Buchholz and Marsi, 2006) and (2) the implementation of the rules to identify the sentence roots.

3.1. Conversion into CoNLL-U

As discussed in Section 2.1.1, words are the basic unit of annotation in the YCOE treebank. They appear between brackets that only contain the part-of-speech tag and the form in which they appear in the sentence. Given these premises, the identification of the tokens to include in the CoNLL-U converted file is almost straightforward. However, it’s worth noting that information such as document and sentence identifiers also appears in the same format as words. For this reason, we had to filter the extracted tokens. To do so, we listed all the possible part-of-speech tags assignable to tokens and we included as tokens only the elements which had one of the tags in the list. We used the list as a table of conversion of the POS tags in the YCOE to a combination of Universal part-of-speech tags and features. In addition to that, we converted the characters such as *thorn*, *eth*, and *ash*, which appeared in their Helsinki equivalents (+t, +d, and +a, and the respective capital letters), to Unicode characters (þ, ð, æ, and the respective capital letters). Doing so, we obtained a CoNLL-U file in which this information was automatically retrieved from the YCOE treebank:

- the sentence id
- the text of the sentence
- for each token:
  - the word form
  - the universal part-of-speech tag
  - the features[6]

[4] The code and data used for this work can be found at https://github.com/unipv-larl/wundorsmitha-geweorc/tree/main/paper_projects/root-identification-oe

[5] *Adrian and Ritheus* is a dialogue on several biblical issues (Cross and Hill, 1982: 3-4). On the other hand, Ælfric’s homily, *Nativitas Domini*, is a Christmas homily, with several expansions, consisting in scriptural elaborations (Pope, 1968: 191-195).

[6] The table used to convert YCOE tags to UD parts-of-speech tags and UD features can be consulted here: https://github.com/unipv-larl/wundorsmitha-geweorc/blob/main/paper_projects/root-identification-oe/pos_table.tsv.

3.2. Rules to identify the roots

The main goal of this work was to define a set of rules that allow to automatically identify the root of the dependency tree, given the annotation of a constituency tree of the OE sentence. In this section, we describe the rules that we implemented.

To define the rules, we benefit from the annotation of the YCOE treebank, which, despite not following a dependency formalism, still gave us some useful information about the syntactic structure of the sentences. In UD, a sentence's root must be unique, and this role can be attributed to tokens with a limited set of features. For example, most of the times adpositions and conjunctions cannot serve as the root of a sentence, but nouns and verbs are eligible for this role. Therefore, having part-of-speech annotation was particularly beneficial in identifying a pool of candidates from which to select the root.

The first step to select the set of candidates, before looking at the parts-of-speech, consist in restricting the number of eligible tokens to those that occupy a relevant position in the constituency tree. The format of each sentence involves a top-level clause that includes isolated tokens and phrases. We focused on the set of isolated tokens (not including punctuation) and we applied some rules taking this set as starting point.

As a matter of example, considering the sentence in Figure 1, we can see that the top-level clause is tagged with the `IP-MAT` label and involves a noun phrase (`NP-NOM`), an adverbial temporal phrase (`ADVP-TMP`), a *that-clause* (`CP-THT`) and an isolated token (`oncneowon`).

The general approach of the procedure to identify the root is exemplified in the algorithm in Figure 2.

In the following sections, we will discuss more in detail each one of the rules mentioned in the algorithm. We will start from the rules that can be applied when the set of isolated tokens is not empty (`VB`, `BE_INF`, `BE_COPULA`, `HAVE`, `BE_ROOT`, `MD`) and then we will move to the other rules (`IP-MAT-0`, `CP_QUE`, `COORD_VB`).

3.2.1. Rule VB

This rule requires a set of isolated tokens to be applied. It considers as good candidates to represent the root of the sentence the isolated tokens whose tag that starts with `VB`, denoting verbs other than the verb 'to be', the verb 'to have' and modal verbs. This rule succeeds in finding the root only if the set of candidates includes one and only one token matching the condition described. It is worth noticing that the verbs might also be tagged with a label preceding `VB`, such as `RP` and `NEG`, denoting the fact that to such verb an adverbial or negative particle is added. So, the `VB` rule assigns the label `root` to the verb.

```
Require: isolated tokens
if not isolated tokens then
    apply IP-MAT-0 rule
    if not root found then
        apply CP-QUE rule
    end if
end if
for all rule in (VB, BE_INF, BE_COPULA, HAVE, BE_ROOT, MD) do
    apply rule on isolated tokens // The rules are applied in the order in which they appear in the list
    if root found then
        break
    end if
end for
if not root found then
    apply CP_QUE
end if
if not root found then
    apply COORD_VB
end if
```

Figure 2: Procedure to identify the root.

```
Require: isolated tokens whose tag starts with BE

if isolated tokens contains one element then
    look for the token following the verb
    if tag following token starts with TO then
        assign root to the verb 'to be'
    end if
end if
```

Figure 3: `BE_INF` rule.

3.2.2. Rule BE_INF

This rule aims to identify all the instances of the verb 'to be'[7] that are parent of an infinitive phrase. To do so, we first look for the isolated tokens which have the tag starting with `BE`; then, if this set consists of only one element, we extract its subsequent element: if its tag starts with `TO`, then we can assign the `root` label to the verb 'to be', as exemplified in Figure 3.

3.2.3. Rule BE_COPULA

This rule aims to find the root in all the situations in which the verb 'to be' acts as a copula of a nominal predicate. According to the UD guidelines, in sentences like these, the root should be assigned to the noun (or adjective) that is the head of the noun (or adjectival) phrase having the role of nominal

[7] Note that the tags starting with `BE` indicate both forms of the two verbs meaning 'to be' (*beon* and *wesan*), but also the forms of the verb *weorþan* 'to become', since it is used as auxiliary to form the passive, or in copular constructions.

```
Require: isolated tokens whose tag starts with BE
if isolated tokens contains one element then
    look for the isolated phrases whose tag ends with NOM-PRD
    if the set of NOM-PRD consists of one element then
        assign root to the head of the NOM-PRD phrase
    end if
end if
```

Figure 4: BE_COPULA rule.

predicate.

We followed the steps as described in Figure 4. We started, as for the BE_INF rule (described in Section 3.2.2), looking for the isolated tokens which have the tag starting with BE; then we looked for the phrases, at the same hierarchical level of the verb 'to be', whose tag ended with NOM-PRD. These combination of labels is used in the YCOE treebank to tag all the predicates in the nominative case. After finding the phrase and checking for its uniqueness in the sentence, we assigned the root label to the head of the noun or adjectival phrases in the predicate.

3.2.4. Rules HAVE, BE_ROOT and MD

The remaining rules can be described as the VB rule in Section 3.2.1. The reason why we differentiate these three rules from the others is that we need to check other conditions before assigning the root label to the verbs 'to have', 'to be', and modal verbs. These three classes of verbs can function as the roots of a sentence, but this happens only under specific conditions (e.g., when nominal predicates, passive verbs, or other finite verbs are not present in the sentence).

These rules are applied at the end, after all the other rules have failed, and they assign the root label to the isolated verbs 'to have', 'to be', and modal verbs, respectively.

3.2.5. Rule IP-MAT-0 and CP_QUE

These rules aims to find the root when the set of isolated tokens is empty. When this happens, we first look for the presence of a phrase whose tag is IP-MAT-0. The -0 tag is used in the YCOE treebank to label all the incomplete IPs (e.g. IPs arisen from elision). Then, after finding the target phrase, we performed the operations described from line 7 to line 12 of the algorithm in Figure 2.

In case of unsuccessful application of the IP-MAT-0 rule, we looked for a phrase whose tag starts with CP-QUE. The type of phrases that match this condition in the YCOE treebank are questions, either indirect or direct (with the addition of the label -SPE). If we found a unique phrase matching the condition, we looked for the presence of a finite subordinate clause (tagged as IP-SUB or IP-SUB-SPE in case of direct speech). Then, as in the previous rule, we applied the rules described in Sections from 3.2.1 to 3.2.4.

3.2.6. Rule COORD_VB

We describe here the last rule we designed, which is aimed at determining the root in sentences where two coordinated elements could potentially both be assigned the root role. We only focused on the situation in which the two coordinates were verbs other than 'to have', 'to be' or modals. In these cases, the extraction of isolated tokens resulted in an empty set (or a set consisting of elements which could not be the root of a sentence). We then looked for a phrase whose tag starts with VB and within that phrase we assigned the root label to the first coordinate, as per the UD guidelines.

The application of these rules didn't always yield the correct root. In certain instances, we were unable to identify a root. In Section 4, we present the outcomes and analyze specific cases.

4. Results

In this section, we describe the results obtained by parsing the YCOE treebank following the rules described in Section 3.

In our study, we analyzed a sample of 390 sentences from Old English texts to assess the performance of our rule-based algorithm. Our objective was to identify the root of each sentence accurately and assign the appropriate label.

correct	wrong	missing	total
349	24	17	390

Table 1: Results of the rule-based root identification.

As Table 1 shows, in 349 out of 390 sentences, following our rule-based approach, we were able to identify the root of the dependency trees correctly. For 24 sentences the root label was assigned to the wrong token, while the 17 cases of 'missing root' were the ones that did not fall in any situation described in our set of rules. Compared to the results obtained by Brigada Villa and Giarda (2023), we can see that the rule-based approach described in this paper reached far better results considering only the root dependency relation (89.49% vs. 78.46%).[8]

[8] The comparison was made replicating the steps described in the GitHub repository of the paper: https://github.com/unipv-larl/wundorsmitha-geweorc/tree/main/paper_projects/parsing_oe_modern.

5. Discussion

In this section, we will analyze the errors made by the model, first addressing the missing roots, i.e. where the model did not succeed in assigning the `root` to any token, and then discussing the wrong roots.

Concerning missing roots, the high majority of them consists in sentence fully or partially in Latin, in which the root is a Latin word. This happens because Latin words are tagged as `FW` in the YCOE, regardless of their actual POS. An example of it is sentence coaelhom,ÆHom_1:23.11 in Figure 5.

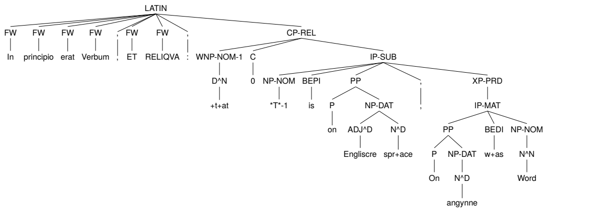

LATIN
FW FW FW FW , FW FW ; CP-REL
In principio erat Verbum , ET RELIQVA : WNP-NOM-1 C IP-SUB
D^N 0 NP-NOM BEPI PP , XP-PRD
+t+at *T*-1 is P NP-DAT , IP-MAT
on ADJ^D N^D PP BEDI NP-NOM
Engliscre spr+ace P NP-DAT w+as N^N
On N^D Word
angynne

Figure 5: Tree of the sentence `coaelhom,ÆHom_1:23.11` whose translation is '*In principio erat Verbum, et reliqua*: that is in the English language "At the beginning there was the Word"'

This sentence comes from a homily, in which the author provides a biblical verse in Latin, immediately followed by its translation in Old English. Despite the presence of Old English words, the root of this sentence is in the Latin part. Out of the 17 missing roots, 10 of them are in Latin sentences. The rest of the sentences are nominal ones, e.g. *& eft þurh Adam on his forgægednysse.* 'And again through Adam in his transgression.' (coaelhom,ÆHom_1:189.109_ID).

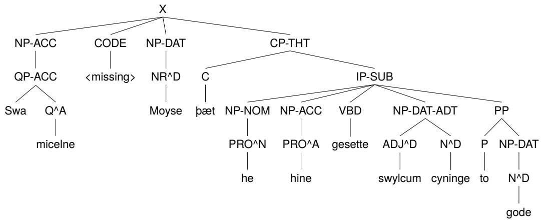

X
NP-ACC CODE NP-DAT CP-THT
QP-ACC <missing> NR^D C IP-SUB
Swa Q^A Moyse þæt NP-NOM NP-ACC VBD NP-DAT-ADT PP
micelne PRO^N PRO^A gesette ADJ^D N^D P NP-DAT
he hine swylcum cyninge to N^D
gode

Figure 6: Tree of the sentence `coaelhom,ÆHom_1:370.193` whose translation is 'so much [...] to Moyses, that he had appointed him god of such king (...)'

Some exceptions to this generalization are, for example, sentence coaelhom,ÆHom_1:370.193 (Figure 6), which contains some missing fragments, or sentence coaelhom,ÆHom_1:41.25 ((Figure 7)), which has the structure of a subordinate clause, introduced by *ac þæt* 'but that', which was not united to the previous one due to the period ending the preceding sentence. In this latter case, in which the sentence starts with a subordinator, but without a main clause, an ad-hoc rule could be implemented to enhance the results of the conversion.

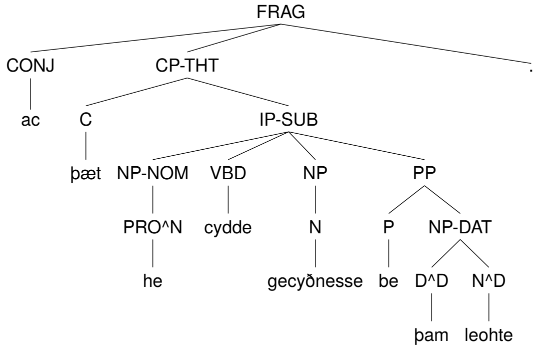


Figure 7: Tree of the sentence `coaelhom,ÆHom_1:41.25` whose translation is 'But so that he announced the witness of the light.'

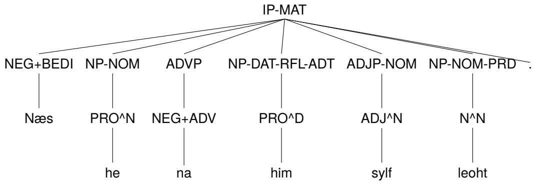

IP-MAT
NEG+BEDI NP-NOM ADVP NP-DAT-RFL-ADT ADJP-NOM NP-NOM-PRD .
Næs PRO^N NEG+ADV PRO^D ADJ^N N^N
he na him sylf leoht

Figure 8: Tree of the sentence `coaelhom,ÆHom1_1:41.24` whose translation is 'He himself is not light'.

As far as wrong attribution of the root is concerned, most of them are connected to the difficulty to discern between copular and existential `BE`. In some cases, only the broader context allows one or the other interpretation. Other errors are linked to the fact that some sentences have a nominal main clause, followed by some subordinates. An example worth discussing is the following: in sentence coaelhom,ÆHom1_1:41.24, *Næs he na him sylf leoht* (Figure 8), the negated verb *nisan* 'not to be' is not recognized as a copula because its YCOE tag was `NEG+BEDI`. This happens because, differently from the `VB`, `HAVE`, `BE_ROOT` and `MD` rules, we could not add the `NEG+` tag to the `BE_COPULA` rule, since it could have been confused with a previous rule and hinder the correct recognition of it.

One last point worth mentioning, is that in sentences such as coaelhom,ÆHom_1:364.192, *Nu ic þe sette, cwæð God sylf to him, þæt þu beo [text missing] Pharaones god [...]* (Figure 9), in which a speech verb interrupts the reported speech content, the model correctly recognizes the verb in the direct speech *sette* 'establish' as the root. The fact that YCOE annotates the interruption as `-PRN`, i.e. appositive or parenthetical, constitutes easy material for the further steps of the conversion since also UD considers these cases as parenthetical parataxis.[9]

[9] `https://universaldependencies.org/u/`

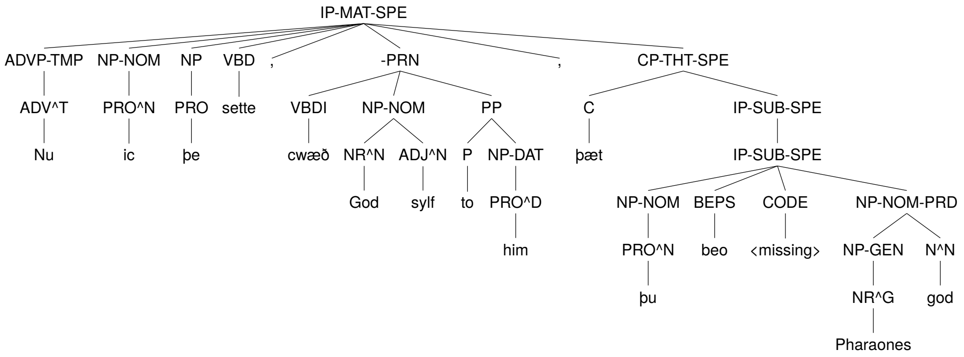


Figure 9: Tree of the sentence coaelhom, ÆHom_1:364.192 whose translation is 'Now I establish for you - said God himself to him, that you be [text missing] the God of the Pharaon [...]'.

6. Conclusions

This paper is the first step towards the creation of a UD treebank for Old English through an automatic conversion of the YCOE treebank from its original constituency format. Since the root is the node from which every other depends, we started with a root-identification task, in which we defined a set of rules to automatically identify the root of a dependency tree, starting from the original YCOE constituency annotation. Given that UD allows only some word classes as roots, we used the original YCOE POS tags as the basis of our rules. After describing Old English morpho-syntax (section 2), we presented, in section 3, our dataset, consisting of manually annotated sentences, and the rules we implemented: section 3.2.1 deals with rule VB, sections 3.2.2 and 3.2.3 present rules concerning the verb 'to be' (BE_INF and BE_COPULA). Rules HAVE, BE_ROOT and MD, described in section 3.2.4, concern verbs which are generally used as auxiliaries, but can nonetheless be the root of a sentence in their lexical meaning. Finally, we presented rules IP-MAT-0 and CP-QUE in section 3.2.5, and rule COORD_VB in section 3.2.6, used when the set of isolated tokens is empty. Our results, discussed in sections 4 and 5, show a precision of 89,23%, thus showing a better performance than a multilingual parser (Brigada Villa and Giarda, 2023). Error analysis has demonstrated that the main errors are due to three factors: a) the presence of Latin sentences; b) the presence of nominal sentences; and c) the difficulty in the disambiguation of copular and existential uses of the verb 'to be'. To conclude, this paper represent a first attempt towards an automatic rule-based conversion of the YCOE annotation into the UD roots and the first step towards the conversion of the whole treebank. The errors analysis may provide a starting point for the implementation of the rules. The use of parsing models for Latin can be used to parse Latin sentences included in the Old English text, in order to have a correct annotation of both languages.

Acknowledgements

We would like to express our gratitude to three anonymous reviewers, whose comments have greatly contributed to improve this paper. This article results from the joint work of the authors. For academic purposes, Martina Giarda is responsible for the manual annotation of the sentences and for Sections 1, 2 and 5 and Luca Brigada Villa is responsible for the code and Sections 3, 4 and 6.

Bibliographical References

Javier Arista. 2022a. Old english universal dependencies: Categories, functions and specific fields. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 3: ICAART,*, pages 945–951. INSTICC, SciTePress.

Javier Arista. 2022b. Toward the morpho-syntactic annotation of an old english corpus with universal dependencies. *Revista de Linguistica y Lenguas Aplicadas*, 17:85–97.

Þórunn Arnardóttir, Hinrik Hafsteinsson, Einar Freyr Sigurðsson, Kristín Bjarnadóttir, Anton Karl Ingason, Hildur Jónsdóttir, and Steinþór Steingrímsson. 2020. A Universal Dependencies conversion pipeline for a Penn-format constituency treebank. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 16–25, Barcelona, Spain (Online). Association for Computational Linguistics.

Luca Brigada Villa and Martina Giarda. 2023. Using modern languages to parse ancient ones: a test on Old English. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 30–41, Dubrovnik, Croatia. Association for Computational Linguistics.

dep/parataxis.html#reported-speech

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.

James E. Cross and Thomas D. Hill. 1982. *The Prose Solomon and Saturn and Adrian and Ritheus*. University of Toronto Press, Toronto.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Olga Fischer, Ans van Kemenade, Willem Koopman, and Wim van der Wurff. 2001. *The Syntax of Early English*. Cambridge Syntax Guides. Cambridge University Press.

Anton Karl Ingason, Hrafn Loftsson, Eiríkur Rögnvaldsson, Einar Freyr Sigurðsson, and Joel C. Wallenberg. 2014. Rapid deployment of phrase structure parsing for related languages: A case study of Insular Scandinavian. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 91–95, Reykjavik, Iceland. European Language Resources Association (ELRA).

Bruce Mitchell and Fred C. Robinson. 2012. *A guide to Old English. Eighth edition*. John Wiley & Sons, Malden, Oxford.

Rafał Molencki. 2017. Syntax. In Laurel J. Brinton and Alexander Bergs, editors, *The History of English. Old English*, volume 2, chapter 5, pages 100–124. De Gruyter Mouton, Berlin, Boston.

Susan Pintzuk and Ann Taylor. 2006. *The Loss of OV Order in the History of English*, chapter 11. John Wiley Sons, Ltd.

John C. Pope. 1968. *Homilies of Ælfric: a Supplementary Collection*. Early English Society, Oxford University Press, London.

Eiríkur Rögnvaldsson, Anton Karl Ingason, Einar Freyr Sigurðsson, and Joel Wallenberg. 2012. The Icelandic parsed historical corpus (IcePaHC). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1977–1984, Istanbul, Turkey. European Language Resources Association (ELRA).

Ferdinand von Mengden. 2017a. Morphology. In Laurel J. Brinton and Alexander Bergs, editors, *The History of English. Old English*, volume 2, chapter 5, pages 73–99. De Gruyter Mouton, Berlin, Boston.

Ferdinand von Mengden. 2017b. Old english: Overview. In Laurel J. Brinton and Alexander Bergs, editors, *The History of English. Old English*, volume 2, chapter 3, pages 32–49. De Gruyter Mouton, Berlin, Boston.

Language Resource References

Kroch, Anthony and Taylor, Ann. 2000. *Penn Helsinki Parsed Corpus of Middle English.* Department of Linguistics, University of Pennsylvania., second. [link].

Pintzuk, Susan and Plug, Leendert. 2002. *The York-Helsinki Parsed Corpus of Old English Poetry (YCOEP)*. Department of Linguistics, University of York. [link].

Taylor, Ann and Warner, Anthony and Pintzuk, Susan and Beths, Frank. 2003. *The York-Toronto-Helsinki Parsed Corpus of Old English Prose (YCOE).* Department of Linguistics, University of York. [link].

Too Young to NER: Improving Entity Recognition on Dutch Historical Documents

Vera Provatorova,[†,⊕] **Marieke van Erp,**[†] **and Evangelos Kanoulas**[⊕]

[†]DHLab, KNAW Humanities Cluster
Oudezijds Achterburgwal 185
1012 DK Amsterdam
The Netherlands
{vera.provatorova,marieke.van.erp}@dh.huc.knaw.nl

[⊕]University of Amsterdam
Science Park 904
1098 XH Amsterdam
The Netherlands
e.kanoulas@uva.nl

Abstract

Named entity recognition (NER) on historical texts is beneficial for the field of digital humanities, as it allows to easily search for the names of people, places and other entities in digitised archives. While the task of historical NER in different languages has been gaining popularity in recent years, Dutch historical NER remains an underexplored topic. Using a recently released historical dataset from the Dutch Language Institute, we train three BERT-based models and analyse the errors to identify main challenges. All three models outperform a contemporary multilingual baseline by a large margin on historical test data.

Keywords: named entity recognition, digital humanities, historical texts

1. Introduction

Named Entity Recognition (NER) is the task of detecting named entities (people, locations, organisations, etc.) mentioned in text (Sang and De Meulder, 2003). NER is widely used for a range of downstream tasks in various domains, including question answering, content recommendation, conversational search and other tasks.

Making digital archives easily searchable is important for researchers in digital humanities, for example for prosopographical research (Tamper et al., 2019). A reliable NER system contributes greatly to this goal: it allows to save manual efforts in looking for information about particular people, places and other entities. However, recognising entities in historical documents is far from a straightforward task: the nature of the data leads to multiple challenges, including OCR noise, historical spelling variations, and potential differences in language use compared to modern texts. The task becomes even more challenging when the documents are written in a low- or mid-resource language: while a vast amount of training data is available for English or French, other languages are less common, leading to a relative lack of parametric knowledge.

While recent advances have been made in recognising and linking historical entities in multiple languages (Ehrmann et al., 2020, 2022), Dutch historical documents remain an underexplored domain, despite the data being publicly available (Dutch Language Institute, 2022). In this paper, we delve into Dutch historial named entity recognition; we train and test three different NER models on historical data ranging from the 17th to the 19th century and provide an extensive analysis of the performance of these models. We hope to inspire further research on Dutch historical NER and draw attention of the research community to the available language resources.

The remainder of this paper is organised as follows. In Section 2, we discuss related work in historical named entity recognition. In Section 3 we detail our experimental setup. We present our results and discussion in Section 4 and conclusions and future work are presented in Section 5. Our code is available at `https://github.com/vera-pro/Dutch-NER-LT4HALA`.

2. Related Work

Languages change over time. In particular prior to the introduction of the printing press and language standardisation language, spelling and writing style variation was widespread. Furthermore, the concepts covered in texts over longer periods of time evolve too, making the analysis and interpretation of historical texts an even greater challenge than contemporary texts (Montanelli and Periti, 2023).

Dutch is a West-Germanic language mainly spoken in the Netherlands, Belgium and Suriname. The language is similar in German in that noun compounding is productive and compounds are generally written without spaces. A term such as notarial deed, made up of ’notary’ and ’akte’ would thus become ’notarisakte’. The language has many loanwords from French, German and Latin. A particular peculiarity that affects named entity recognition is that it is common for family names to contain location names (Brouwer et al., 2022). Prior to the 18th century, there was no standard Dutch spelling. Although various attempts were made to establish

dataset	century span	# entity annotations			data source
		PER	LOC	TIME	
train	17th-19th	55,921	30,636	19,809	see test: SA, test: VOC, test: RHC, test: NHA
validation	17th-19th	14,393	7,427	4,782	see test: SA, test: VOC, test: RHC, test: NHA
test: SA	17th-18th	781	257	255	Notarial deeds from the Amsterdam City Archive
test: VOC	17th-18th	290	315	180	Notarial deeds of the Dutch East India Company
test: RHC	19th	24	17	5	Notarial deeds from the archives of the Dutch regional historic centra
test: NHA	19th	352	252	109	Notarial deeds archive of Haarlem
test: CoNLL'02	21st	1098	774	0	Belgian newspaper "De Morgen" of 2000 (editions from June to September)

Table 1: Dataset details. The training and validation splits, as well as historical test splits, are part of (Dutch Language Institute, 2022). The contemporary test set is from (Tjong Kim Sang, 2002).

a guide, none gained widespread adoption. With the rise of printing, spelling standardization accelerated. Modern Dutch spelling can be traced back to the 1860s, when Matthijs de Vries and Lammert Allard proposed a set of spelling rules and word lists forming the basis of contemporary written. These efforts were supported by the government (Donaldson, 1983).[1]

Contemporary language models such as BERT (Devlin et al., 2019), Bloom (Scao et al., 2022) and LLaMA (Touvron et al., 2023) are optimised for contemporary language. This means these models may not perform as well on historical texts that differ from modern language (Hosseini et al., 2021; Lai et al., 2021). Historical texts often contain obsolete expressions or words with different meanings than today. Additionally, spelling variations and OCR errors may limit the accuracy of automated text processing systems.

The task of historical NER has been gaining popularity in the recent years, with domain-specific NER research focusing on for example medieval Latin charters (Chastang et al., 2021) or historical locations (Won et al., 2018). (Ehrmann et al., 2020) introduced HIPE, a shared task focused on recognising and linking entities in historical newspapers. Two years later, the next shared task on this topic has been introduced by the same team (Ehrmann et al., 2022). The languages in HIPE '20 include English, German and French, with Finnish and Swedish added as extra languages in HIPE '22.

The contributions most similar to ours are (Hendriks et al., 2020), where the authors performed NER and record linkage on historical Amsterdam notarial archives and personnel records of the United East Indies Company (VOC), and (Arnoult et al., 2021), where the authors experimented with Dutch and multilingual NER models on their new dataset of VOC records. As this work was done prior to the latest iteration of LLMs and the introduction of the NER dataset by the Dutch Language Institute, we further build upon and extend the understanding of NER performance on historical Dutch texts. For further reading, we refer the reader to the following historical NER surveys: (Blouin et al., 2021; Humbel et al., 2021; Ehrmann et al., 2023).

3. Experimental Setup

Following (Sang and De Meulder, 2003), we approach NER as a token classification problem. We focus on transformer-based models as these provide the best performance and ease of use in transfer learning at the time of writing (Li et al., 2020). In this section, we detail which models were used and how we fine-tuned them, the datasets we tested on, and the approach we used for evaluation and error analysis.

3.1. Models

We fine-tune three BERT-based models on historical data:

1. BERTje (De Vries et al., 2019), a Dutch model trained on a mixture of modern texts and historical novels, with modern texts being the majority in the training data;
2. GysBERT (Manjavacas and Fonteyn, 2022), a Dutch model designed specifically for historical data;
3. mBERT (Devlin et al., 2019), a multilingual model that includes Dutch as one of its languages.

The models were trained on one GPU for 15 epochs with early stopping. We used the batch size 8 and selected the best checkpoint by F1 score. To evaluate the models against a strong baseline that has not been optimised for historical data, we compare them with WikiNEuRal (Tedeschi et al., 2021). This

[1] https://www.dbnl.org/tekst/dona001dutc02_01/dona001dutc02_01_0007.php

is a multilingual NER model that includes Dutch as one of its languages and achieves high scores on contemporary benchmarks.

3.2. Datasets

We fine-tune the models using the training and validation splits of the NER dataset provided by Dutch Language Institute (2022). This dataset was created in 2020 through a crowdsourcing project initiated by the Dutch National Archive. The dataset contains notarial deeds from eleven different Dutch archives, some focused on Dutch East India Company dealings, others on local notary business. For testing the models, we use the test splits of Dutch Language Institute (2022) as well as a dataset with modern texts: the test split of Tjong Kim Sang (2002). Table 1 shows the details of the datasets. There are many different NER categorisations. In (Dutch Language Institute, 2022) the labels PER, LOC and TIME are present, while for (Tjong Kim Sang, 2002) the labels are PER, LOC, ORG, and MISC. Since the last two labels are not seen by the models in the training data, we exclude them from evaluation. As WikiNEuRal has extra NER labels in its vocabulary, we consider the predictions containing these labels as 'O' when comparing the models.

3.3. Evaluation

To identify main challenges in historical Dutch NER, we first group the data subsets by century to analyse the role of time. We analyse precision and recall of the models per century, create confusion matrices, identify overlaps in the wrong predictions made by different models, and perform qualitative analysis to find examples of challenging NER cases.

4. Results and Discussion

This section describes the results of our experiments and the error analysis. Table 2 shows precision, recall and F1 score per model per century for two NER labels, PER and LOC (TIME is excluded from this part of the analysis since WikiNEuRal does not predict it). For both labels the same pattern is observed: WikiNEuRal achieves best results on contemporary data and performs substantially worse than all other models on historical data. Interestingly, GysBERT does not outperform BERTje and mBERT on historical data, despite having seen more historical texts during pre-training: the three models achieve approximately the same results. On the contemporary test set, however, mBERT performs worse than all other models, achieving particularly low scores in both precision and recall on the LOC entity class.

Figure 1 shows confusion matrices for all labels per model per century. The main diagonal displays the number of correctly classified tokens for each label. Note that the exact number of tokens may vary per model, since each model has its own WordPiece tokenizer. From the figure we identify four most common classes of errors:

1. "False positive": predicting an entity when the correct label is "O";
2. "False negative": predicting "O" when the correct label is an entity;
3. Mention boundaries: predicting a correct class but with "I-" instead of "B-" and vice versa;
4. People vs. places: confusing "PER" and "LOC" entities.

When looking closely at the error examples during our qualitative evaluation, we noticed that some errors are caused by wrong annotations in the test sets: for example, the entity "Willem van Zonneveld" in the NHA test set is labelled as two separate PER entities, "Willem van" and "Zonneveld", which is incorrect. All models except WikiNEuRal recognise this entity correctly, which leads to a mention boundaries error. Some errors, however, are indeed caused by the models making wrong predictions: for example, in the CoNLL test set mBERT incorrectly predicts two separate LOC entities for "Los Angeles". In case of the "people vs. places" errors, qualitative analysis shows that many examples are ambiguous, and some of the mistakes made by the models could be also made by a human annotator. For example, "Jan Hendrik du Caijlar van Delf" in the VOC test set is labelled as one PER entity with a double surname, but all models predict "Delf" as a separate entity, as in "Jan Hendrik du Caijlar from Delft". This type of errors is an interesting challenge typical for Dutch texts, since Dutch family names often contain location names (Brouwer et al., 2022).

Figure 2 is a Venn diagram showing the overlap in wrong predictions between models for every test set. Note that an overlap between two models here means that both models gave a wrong answer, but the answer is not necessarily the same for the two models. The error overlap is small for all historical test sets, which indicates that the models tend to make different mistakes and therefore could benefit from ensembling.

5. Conclusion and Future Work

We used historical texts from the Dutch Language Institute to train three BERT-based NER models, making one of the first steps towards publicly available Dutch historical NER. All models are shown to

label	model	century								
		17-18			19			20		
		P	R	F	P	R	F	P	R	F
PER	GysBERT	.71	.67	.69	.76	**.73**	.74	.74	.76	.75
	BERTje	**.76**	**.71**	**.73**	**.80**	**.73**	**.76**	.88	.83	.85
	mBERT	.72	.68	.70	.77	.72	.74	.74	.71	.72
	WikiNEuRal	.48	.40	.43	.61	.45	.51	**.94**	**.86**	**.90**
LOC	GysBERT	.74	**.79**	.76	**.81**	**.77**	**.79**	**.72**	.66	.69
	BERTje	.77	.78	**.78**	.78	**.77**	.78	.71	.71	.71
	mBERT	**.79**	.77	**.78**	**.81**	.75	.78	.51	.48	.50
	WikiNEuRal	.48	.50	.49	.50	.48	.49	**.72**	**.90**	**.80**

Table 2: Precision, recall and F1 score per century on the PER and LOC labels.

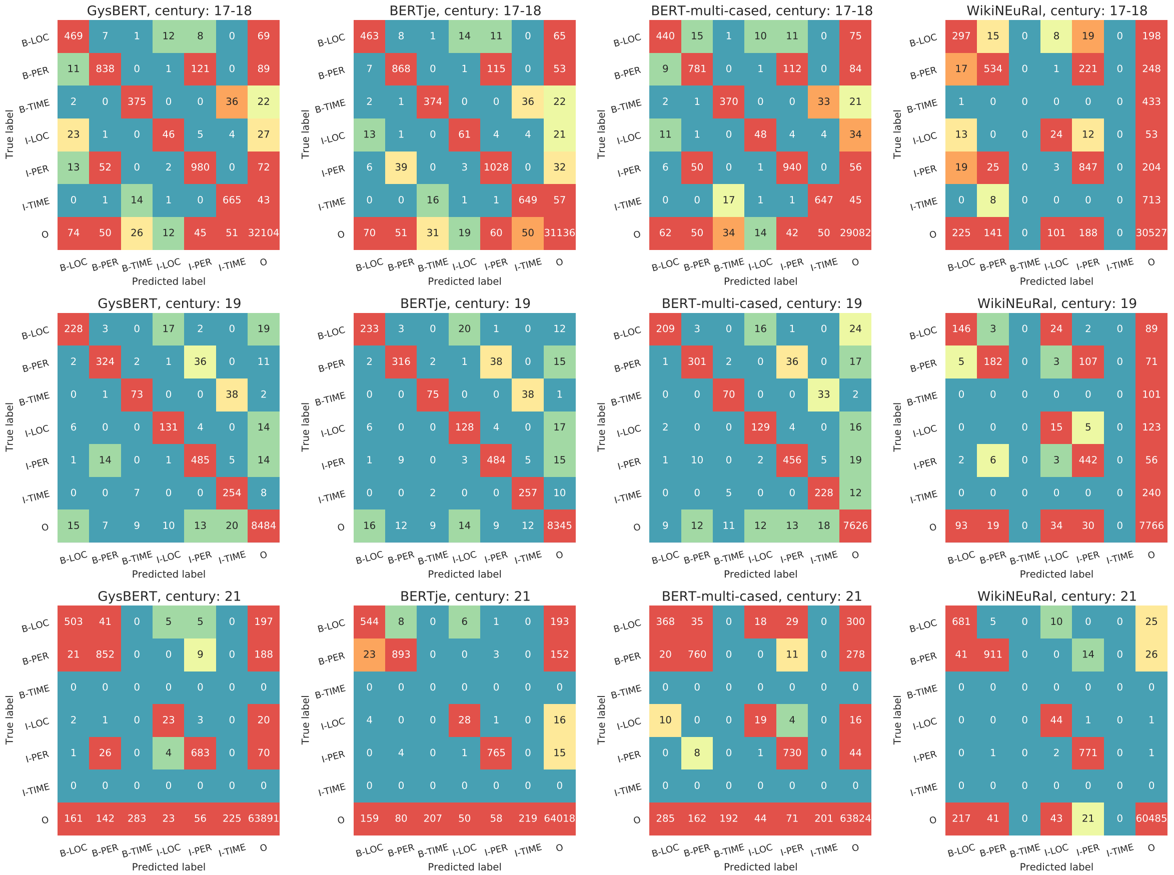


Figure 1: Confusion matrices of the models per token per century. Every cell shows a number of tokens.

perform well on historical data from the 17th to the 19th century, achieving substantially better scores than the baseline. Our error analysis shows that the overlap in wrong predictions on historical data is small, which indicates that using an ensemble of the three models might be optimal for recognising entities in Dutch historical data. Future work includes implementing and testing such an ensemble, as well as experimenting with more diverse entity types and testing on additional domains.

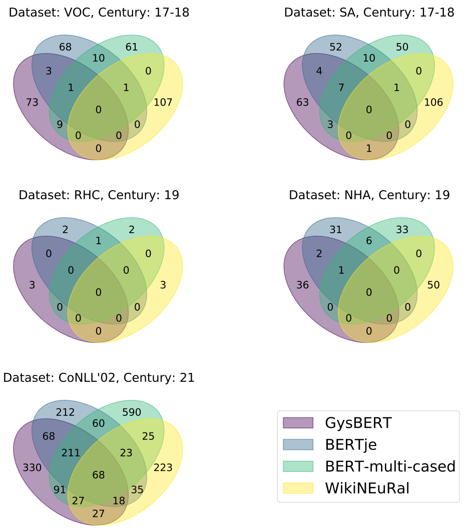


Figure 2: The overlap of false predictions per dataset. Every petal shows a number of sentences with at least one wrong prediction.

6. Acknowledgements

This research was supported by the KB National Library of the Netherlands Researcher-in-Residence program, NWO Smart Culture – Big Data / Digital Humanities (314-99-301), the Informatics Institute of the University of Amsterdam, and the European Union (grant agreement 101088548 - TRIFECTA). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. We thank Sara Veldhoen, Marieke Moolenaar and Willem Jan Faber for their helpful feedback.

7. Bibliographical References

Sophie I Arnoult, Lodewijk Petram, and Piek Vossen. 2021. Batavia asked for advice. Pretrained language models for named entity recognition in historical texts. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 21–30.

Baptiste Blouin, Benoit Favre, Jeremy Auguste, and Christian Henriot. 2021. Transferring modern named entity recognition to the historical domain: How to take the step? In *Proceedings of the Workshop on Natural Language Processing for Digital Humanities*, pages 152–162.

Leendert Brouwer, Peter McClure, and Charles Gehring. 2022. Dutch family names. In *Dictionary of American Family Names*. Oxford University Press.

Pierre Chastang, Sergio Torres Aguilar, and Xavier Tannier. 2021. A named entity recognition model for medieval latin charters. *Digital Humanities Quarterly*, 15(4).

Wietse De Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A dutch BERT model. *arXiv preprint arXiv:1912.09582*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bruce Donaldson. 1983. *Dutch: A linguistic history of Holland and Belgium*. Uitgeverij Martinus Nijhoff, Leiden.

Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. 2023. Named entity recognition and classification in historical documents: A survey. *ACM Computing Surveys*, 56(2):1–47.

Maud Ehrmann, Matteo Romanello, Alex Flückiger, and Simon Clematide. 2020. Extended overview of clef hipe 2020: named entity processing on historical newspapers. In *CLEF 2020 Working Notes. Conference and Labs of the Evaluation Forum*, volume 2696. CEUR-WS.

Maud Ehrmann, Matteo Romanello, Sven Najem-Meyer, Antoine Doucet, Simon Clematide, Gulielmo Faggioli, Nicola Ferro, Alan Hanbury, and Martin Potthast. 2022. Extended overview of hipe-2022: Named entity recognition and linking in multilingual historical documents. In *CEUR Workshop Proceedings*, 3180, pages 1038–1063. CEUR-WS.

Barry Hendriks, Paul Groth, and Marieke van Erp. 2020. Recognizing and linking entities in old dutch text: A case study on voc notary records. In *COLCO*, pages 25–36.

Kasra Hosseini, Kaspar Beelen, Giovanni Colavizza, and Mariona Coll Ardanuy. 2021. Neural language models for nineteenth-century english. *Journal of Open Humanities Data*.

Marco Humbel, Julianne Nyhan, Andreas Vlachidis, Kim Sloan, and Alexandra Ortolja-Baird. 2021. Named-entity recognition for early modern textual documents: a review of capabilities and challenges with strategies for the future. *Journal of Documentation*, 77(6):1223–1247.

Viet Lai, Minh Van Nguyen, Heidi Kaufman, and Thien Huu Nguyen. 2021. Event Extraction from Historical Texts: A New Dataset for Black Rebellions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2390–2400, Online. Association for Computational Linguistics.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Enrique Manjavacas and Lauren Fonteyn. 2022. Non-parametric word sense disambiguation for historical languages. In *Proceedings of the 2nd International Workshop on Natural Language Processing for Digital Humanities*, pages 123–134.

Stefano Montanelli and Francesco Periti. 2023. A survey on contextualised semantic shift detection.

Clemens Neudecker. 2016. An open corpus for named entity recognition in historic newspapers. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4348–4352, Portorož, Slovenia. European Language Resources Association (ELRA).

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Minna Tamper, Petri Leskinen, and Eero Hyvönen. 2019. Visualizing and analyzing networks of named entities in biographical dictionaries for digital humanities research. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 199–214. Springer.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021. Wikineural: Combined neural and knowledge-based silver data creation for multilingual ner. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Miguel Won, Patricia Murrieta-Flores, and Bruno Martins. 2018. ensemble named entity recognition (ner): evaluating ner tools in the identification of place names in historical corpora. *Frontiers in Digital Humanities*, 5:2.

8. Language Resource References

Dutch Language Institute. 2022. *AI-Trainingset for NER (Version 1.0)*. Dutch Language Institute. Dutch Language Institute, 1.0. [link].

Tjong Kim Sang, Erik F. 2002. *Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition*. [link].

Towards Named-Entity and Coreference Annotation of the Hebrew Bible

Daniel G. Swanson, Bryce D. Bussert, Francis M. Tyers
Indiana University, Gateway Seminary, Indiana University
Department of Linguistics, Department of Biblical Studies, Department of Linguistics,
Bloomington, Indiana, Ontario, California, Bloomington, Indiana
dangswan@iu.edu, bussertscholar@gmail.com, ftyers@iu.edu

Abstract

Named-entity annotation refers to the process of specifying what real-world (or, at least, external-to-the-text) entities various names and descriptions within a text refer to. Coreference annotation, meanwhile, specifies what context-dependent words or phrases, such as pronouns refer to. This paper describes an ongoing project to apply both of these to the Hebrew Bible, so far covering most of the book of Genesis, fully marking every person, place, object, and point in time which occurs in the text. The annotation process and possible future uses for the data are covered, along with the challenges involved in applying existing annotation guidelines to the Hebrew text.

Keywords: Ancient Hebrew, coreference, named-entity

1. Introduction

Coreference annotation is the process of marking whether or not two words or phrases in a document refer to the same document-external entity (whether real or imagined), which is very useful for information retrieval.

Named-entity and coreference annotation allows scholars, language instructors and students to view and search a version of the text that goes beyond lemmas to represent the real-world entities that tie the text together. Searches in this corpus can provide all mentions to a real-world entity, not only instances of a particular lemma, and facilitate linguistic inquiries at the syntax-semantics interface– where the entity type affects its usage in the sentence.

There are two main ways of accomplishing such annotations: links and clusters (Nedoluzhko et al., 2022). With linked annotations, each marked phrase is attached to another phrase (generally the nearest preceding one) with which it corefers. With clusters, on the other hand, a separate list of entities is created and each phrase is tagged as referring to one of those entities.

This paper presents the creation of a corpus of Ancient Hebrew annotated for co-reference using the cluster method, which thus also serves a set of named-entity annotations as well.

Ancient Hebrew is a Semitic language formerly spoken in the region that is now Israel and Palestine in the first and second millennia BC which survives to the present day in liturgical contexts.

The available corpus of texts in Ancient Hebrew (as distinct from Mishnaic Hebrew, a daughter language used by Jewish scholars during the Middle Ages) consists primarily of versions of the documents that now make up the Hebrew Bible. The standard versions of these texts contain 300-500 thousand words, depending on tokenization.

Lemmatization and part-of-speech tagging for the entirety of this corpus were completed by Peursen et al. (2015) and the first section of the corpus (30 thousand words) were syntactically annotated using the Universal Dependencies framework in Swanson and Tyers (2022). In this paper, we present the results of a pilot study on expanding the Universal Dependencies treebank to include co-reference and named-entity annotations.

The paper is organized as follows: Section 2 discusses the annotation scheme and the tools used in the annotation process. Section 3 provides a variety of statistics concerning the distribution of the resulting annotations. Section 4 describes the steps taken to measure the annotation quality and Section 5 concludes.

2. Annotation

In this project, we followed the CorefUD standard (Nedoluzhko et al., 2022), which is designed to be compatible with the Universal Dependencies file format. This meant that the annotations are done such that each phrase ("mention") points to an entry in a separate list rather than to a preceding (or, perhaps, following) phrase to with which it corefers.

CorefUD does not, however, provide definitions for what should and should not be included in the annotations. For this we used a subset of the co-reference guidelines used in the Universal Dependencies English GUM treebank (Zeldes, 2017), specifically the criteria for being a mention, the list of entity types, and the criteria for identifying two mentions as coreferential[1].

[1] The GUM guidelines can be found at `https://`

Following the definitions in GUM, every noun phrase, proper noun, and pronoun in the corpus (including nested phrases) was included as a mention, apart from interrogative pronouns and a handful of a few language-specific constructions deemed to be non-referential, such as לבדו /levado/ "alone" (literally "to his separation"), where the central element בד /vad/ is a noun and thus forms a noun phrase with the possessive pronoun ו /o/, but the phrase has no meaningful referent. In this instance the pronoun is marked as coreferential with the appropriate entity (usually a person), while the noun is not part of any mention. Demonstrative adverbs such as שׁם /sham/ "there" are also mentions, as are clauses and coordinated noun phrases which are referred back to. A consequence of this is that the resulting list of entities includes things that would not be found in any external ontology, such as the individual animals being sacrificed in a particular passage or an entity for a person's name separate from the entity for the person themself. The latter case occurs several times when describing the birth of a child, where text typically has some variant of "And they called his name 'Isaac'." Here *his* refers to Isaac, while *his name* and *'Isaac'* refer to Isaac's name rather than to Isaac himself.

Each entity is assigned one of the 10 types used in GUM and CorefUD. These are listed in Table 1. The definitions have been retained from GUM, but some names have been changed solely so that no two types have the same first letter, allowing us to use single letter mnemonics in our data files and annotation interface as described below.

The coreference guidelines from GUM which were used in this project primarily pertain to the circumstances under which copular predicates are or are not considered to corefer with their subjects.

To produce the annotations, the rule-based coreferencer Xrenner (Zeldes and Zhang, 2016) was applied to the treebank. The mentions it detected were exported, but our initial investigation found that the accuracy of its coreference labels was too low to be particularly helpful, so we opted to discard these. A simple terminal interface was then constructed in Python which displays a mention and its immediate context to the annotator who can then choose to label it with an existing entity or create a new entity. Entities can be referred to by ID, which consists of the first letter the entity type and a number counting up sequentially from the beginning of the corpus. Thus, when this project is expanded to include the entire UD treebank, the first three Person entities will be God (p1), Adam (p2), and Eve (p3). Many of the entities also have human-readable names, for which the annotation interface provides an autocomplete function (adding a name is optional if the annotator is confident that the entity in question is only referred to once). An example of the interface is given in Figure 1.

All the code used in this project is freely available and can be found with the data at https://github.com/mr-martian/hbo-UD. The data will also be converted to the CorefUD format and included in the upcoming version 1.2 release.

3. Corpus Statistics

The underlying corpus of the present project is a portion of the UD_Ancient_Hebrew-PTNK treebank (Swanson and Tyers, 2022) as of Universal Dependencies version 2.13 (Nivre et al., 2020), specifically containing the test and development sets and half of the training set. The size of this corpus is summarized in Table 2. This comprises the first 40 chapters of Genesis.

The coreference annotations label over 10,000 mentions referring to almost 1500 distinct entities. The distribution of entities and mentions by type is given in Table 3. The most common entity type is Person, which covers roughly 35% of the entities and 70% of the mentions. The least common, meanwhile, is Vegetation, at 1.5% of the entities and 0.6% of the mentions.

Eleven entities are referred to more than 100 times. All but one of them are Persons: The patriarchs Abraham (524), Isaac (208), Jacob (567), and Joseph (173), God (437), Jacob's brother Esau (189), Jacob's uncle Laban (156), Abraham's wife Sarah (122), Isaac's wife Rebecca (113), and one of Abraham's servants (102). The only location with more than 100 mentions is "the world" (149). Together these 11 entities total 2740 mentions, 37% of the total.

At the other end, there are 867 entities which are only mentioned once, which is 58% of all entities and 12% of all mentions.

4. Evaluation

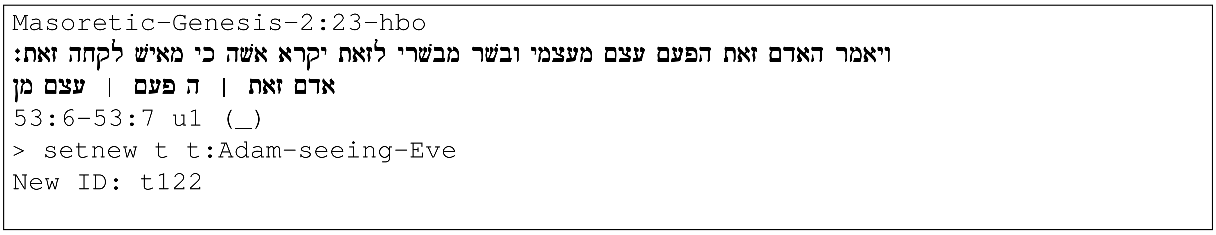
One of the 40 chapters (specifically, Genesis 6) was chosen at random to be annotated twice. We measured agreement using the metrics provided by the corefUD scorer[2], which is an evaluation tool based on the Universal Anaphora Scorer (Yu et al., 2022), but adapted to the corefUD format. Each metric compares a reference document to a system output, so we ran the scorer with each annotator as the reference and averaged the resulting scores. The results are shown in Table 4. In addition, we give an analysis of the raw agreement rates on span selection, coreference, and entity type, since

wiki.gucorpling.org/gum/entities.

[2] https://github.com/ufal/corefud-scorer

GUM Label	Our Label	Examples
person	person	God, Abraham, the messenger of God
place	location	Bethel, Egypt, in the field
organization	nation	the Egyptians, the army of the Philistines
object	inanimate	a water-skin, a gold nose-ring
event	event	a feast, this thing that you have done
time	time	forever, the morning after the feast
substance	substance	the water of the well, the gold of that land
animal	creature	Abraham's donkeys, seven fat cows
plant	vegetation	the Tree of Life, a bush
abstract	abstract	his love for Rachel, favor in your sight

Table 1: The 10 entity types used in the corpus and how they relate to the GUM entity types. The names of the types used in the current corpus were chosen so as to be uniquely identifiable by their first letter.



```
Masoretic-Genesis-2:23-hbo
ויאמר האדם זאת הפעם עצם מעצמי ובשׂר מבשׂרי לזאת יקרא אשה כי מאישׂ לקחה זאת:
אדם זאת | ה פעם | עצם מן
53:6-53:7 u1 (_)
> setnew t t:Adam-seeing-Eve
New ID: t122
```

Figure 1: The interface of the annotation tool. The first line gives the id of the sentence in the treebank. The second gives the full text of the sentence (in this case it reads "And the man said 'This one, now, is bone from my bone and flesh from my flesh. Because of this she shall be called "woman" because from man she was taken.'.") and the third gives the lemmas of each word in the current mention (here הפעם /hapa'am/ "now") along with the nearest two words on either side. The next line is the internal representation of the mention. `53:6-53:7` indicates that the mention begins at the 6th word of sentence 53 and ends at the 7th. `u1` is the current entity associated with this mention, in this case the first unknown and _ is the human-readable name of the entity, which is empty, since this is an unknown. `>` is a prompt for a command and the command here entered assigns this mention to a newly-created Time (`t`) entity with the name "t:Adam-seeing-Eve", which turns out to be the 122nd time entity created in this corpus.

	UD	CorefUD	Used
Sentences	1,579	1,161	73.5%
Words	39,036	28,485	73.0%
Tokens	26,846	19,621	73.1%

Table 2: Statistics about the UD_Ancient_Hebrew-PTNK treebank which was formed the basis of this project as of UDv2.13 and the resulting coreference corpus. The final column gives the proportion of the UD data which was used in the present work.

these 3 areas more directly show ways of improving the annotation process. A summary of these agreement rates is also given in Table 4.

4.1. Span Selection

The automated annotations consisted of 202 spans. Given the actions of 'annotate', 'delete', and 'modify', the two annotators agreed in 179 cases (88.61%). An analysis of the disagreements found that Xrenner overgenerates spans for entity mentions and the annotation guidelines were unclear on the proper treatment of some phenomena.

For example, Xrenner gives some determiners separate mentions due to part-of-speech tags. In (1), the word כל (kol "all, whole") is a noun, both

Entity type	Entity count	Mention count
Person	477	4842
Location	187	833
Abstract	218	429
Inanimate	173	372
Creature	100	276
Nation	73	259
Time	150	227
Substance	40	94
Vegetation	30	72
Event	47	69
Total:	1495	7473

Table 3: The frequency of entities and mentions in the corpus by entity type, sorted by number of mentions.

Measure	Agreement Rate	
Spans	179 / 202	88.61%
Spans (corrected)	188 / 202	93.07%
Coreference	129 / 147	87.76%
Entity Type	121 / 147	82.31%
LEA	70.02	±1.15
MUC	81.44	±1.04
B^3	73.55	±0.69
CEAFe	62.66	±1.11
CEAFm	77.73	±0.87
BLANC	78.32	±0.92
CoNLL	72.55	±0.94

Table 4: Inter-annotator agreement statistics for Genesis chapter 6. "Spans" and "Spans (corrected)" refers to filtering of the original list of spans before and after an automated correction step was added (see Section 4.1). "Coreference" refers to the rate of agreement on which spans are and are not the same entity (Section 4.2). And "Entity Type" refers to whether the types of the entities match (Section4.3). The other scores are the F1 scores reported by the corefUD scorer. The scores are not symmetric with respect to which set of annotations is the reference, so we report the average (with variation) of the two directions.

etymologically and in the UD part-of-speech tags, and thus Xrenner creates mentions for both "the whole land" and "the land", when only the former should be annotated.

(1)	ההוא	הארץ	כל
	ה-הוא	ה-ארץ	כל
	3SG.M-DEF	land-DEF	whole

"the whole of that land"

Similarly, ההוא (hahu' "the-him, that") is the 3rd person singular masculine pronoun with a definite article, a construction which serves as a demonstrative rather than as a referential pronoun. Thus, Xrenner produces a distinct mention for "that" in addition to "that land".

Fortunately, these issues, and a related one for numerals, can be fixed with an automated preprocessing step. Further, they can be automatically filtered from the Xrenner output, thus reducing annotator effort and risk of error.

Automatic correction took care of 9 disagreements, raising the agreement rate for span identification to 188 / 202 (93.07%).

4.2. Coreference

147 spans were given a label by both annotators. We calculate coreference agreement as follows: Given that annotator 1 applied a particular label to a set of spans, how many of those spans did annotator 2 label as coreferential? For example, if annotator 1 assigned a label of `i12` to 5 spans and annotator 2 assigned `s9` to 3 of the same spans and `c4` and `c5` to the other 2, we would calculate the coreference agreement by saying that annotator 2 agrees that 3 / 5 (60%) of spans are coreferential to one another (the particular labels being ignored for this measure).

Using the measure on the test sample, we observe an agreement rate of 129 / 147 (87.76%).

An example of an instance where the annotators disagreed was in Genesis 6:2, which refers בני האלהים /beney ha'elohim/ "the sons of God/the gods". Both annotators agreed on the coreference of the larger phrase as being a mysterious group not mentioned elsewhere, but one interpreted the nested mention as one of the names of God while the other read it as a plural noun referring to some other group of supernatural figures. The released version of the data takes the first interpretation, somewhat arbitrarily, pending a further analysis of evidence beyond the local lexical and syntactic context, since neither of those provide grounds for a decision.

4.3. Entity Types

Of the 147 spans annotated by both annotators, there were 26 cases where the entity type differed between them, giving an agreement rate of 121 / 147 (82.31%). The primary source of disagreement (14 of the 26 differences) was due to an unclear definition of the "nation" ("organization") entity type. It was sometimes used to refer to any group, though the intended use was for a group of people such that changing the specific members does not change the identity of the group (for example, the people of Egypt or the Philistine army). Thus, one annotator marked the set of all humans and animals as "nation" while the other marked it as "creature" (the released data has "creature"). Existing entities of this type have been reviewed and corrected as necessary.

5. Conclusion

In this paper we have presented a corpus of coreference annotations for Ancient Hebrew along with a description of the annotation guidelines and process, and distribution statistics distribution of various features in the text. We also presented the inter-annotator agreement of the text with discussion of methods to increase agreement via clarifications of the guidelines and improvements to the annotation pipeline.

In the future, we plan to expand the corpus to cover the rest of the Hebrew Bible. In addition, there are several other types of annotations which commonly accompany co-reference, such as annotating relationships between entities (e.g. bridging, or part-whole relationships), which can be partially derived from our entity naming process, and linking the entity IDs to external sources, such as Wikipedia. Such extensions would greatly enhance the usefulness of this resource by enabling more complex querying of the data.

Acknowledgements

We would like to think Naomi Brokema and Amir Zeldes for discussing particularly tricky annotations decisions and thus helping to clarify the annotation scheme.

6. Bibliographical References

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, Amir Zeldes, and Daniel Zeman. 2022. CorefUD 1.0: Coreference meets Universal Dependencies. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4859–4872, Marseille, France. European Language Resources Association.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043.

W.T. van Peursen, C. Sikkel, and D. Roorda. 2015. Hebrew text database ETCBC4b.

Daniel Swanson and Francis Tyers. 2022. A Universal Dependencies treebank of Ancient Hebrew. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2353–2361, Marseille, France. European Language Resources Association.

Juntao Yu, Sopan Khosla, Nafise Sadat Moosavi, Silviu Paun, Sameer Pradhan, and Massimo Poesio. 2022. The universal anaphora scorer. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4873–4883, Marseille, France. European Language Resources Association.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Amir Zeldes and Shuo Zhang. 2016. When annotation schemes change rules help: A configurable approach to coreference resolution beyond OntoNotes. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016)*, pages 92–101, San Diego, California. Association for Computational Linguistics.

LiMe: a Latin Corpus of Late Medieval Criminal Sentences

Alessandra Bassani[1], **Beatrice Del Bo**[2], **Alfio Ferrara**[3], **Marta Mangini**[2], **Sergio Picascia**[3], **Ambra Stefanello**[4]

[1]Università degli Studi di Milano, Department of Italian and Supranational Public Law
[2]Università degli Studi di Milano, Department of Historical Studies
[3]Università degli Studi di Milano, Department of Computer Science
[4]Università degli Studi di Firenze, Department of History, Archaeology, Geography, Fine and Performing Arts

Abstract

The Latin language has received attention from the computational linguistics research community, which has built, over the years, several valuable resources, ranging from detailed annotated corpora to sophisticated tools for linguistic analysis. With the recent advent of large language models, researchers have also started developing models capable of generating vector representations of Latin texts. The performances of such models remain behind the ones for modern languages, given the disparity in available data. In this paper, we present the LiMe dataset, a corpus of 325 documents extracted from a series of medieval manuscripts called *Libri sententiarum potestatis Mediolani*, and thoroughly annotated by experts, in order to be employed for masked language model, as well as supervised natural language processing tasks.

Keywords: latin corpus, medieval case law, natural language processing

1. Introduction

The manuscripts called *Libri sententiarum potestatis Mediolani*, preserved at the *Archivio Storico Civico and Biblioteca Trivulziana in Milan, Cimeli, 146-152*, represent all that remains of the documentation recorded in the late medieval period at the court of justice of the city of Milan. The seven manuscripts of the series cover the activity of the court during the years 1385, 1390-1392, 1397-1398, 1398-1399, 1400-1401, 1427 and 1428-1429, respectively, resulting in the delivery of approximately 3,000 criminal sentences[1] discussed in the presence of the Milanese judges, pronounced by the *podestà*[2] and publicly recorded by the notaries who worked at the court in the *Loggia degli Osii*[3]. Although, as evident, the chronological span of each *Liber* varies considerably according to the length of time each *podestà* was in office, the structure, the material aspect and even the form employed in the drafting of these manuscripts present elements of a certain homogeneity and uniformity. This is due to the fact that the notaries in charge of assisting mayors and judges during trials recorded the sentences according to a pattern that is repeated almost unchanged in all manuscripts.

Each verdict, preceded by the verbal invocation - *In nomine Domini, amen*[4] - is pronounced by the *podestà* in accordance with the seigniorial decrees and statutes of the municipality of Milan. It contains the names of the accused, the narration of the legal proceeding, whether it was an *inquisitio* or an accusation, with the salient phases of the trial and the final pronouncement. In addition to the sentences, whose pattern is formally identical for all defendants, there are also numerous subsequent interventions: e.g. annotations relating to receipts for full or partial payment of penalties or cancellations of sentences.

The *Libri sententiarum potestatis Mediolani* are pivotal sources for law historians, like all Medieval and Early Modern trial outcomes preserved in the European archives: they allow us to measure the distance between the discipline established by *statuta* and *ius comune* and its actual application before the courts of medieval cities (Padoa-Schioppa, 2017). Indeed, the seven *Libri* photograph the complex balance of social and political forces that characterised the city of Milan during the Visconti rule (Gamberini, 2014).

This documentary typology constitutes a source of great importance for historians of medieval law (Storti, 2021; Valsecchi, 2021; Bassani, 2021; Isotton, 2021; Bianchi Riva, 2021; Minnucci, 2021), meanwhile fulfilling the same function for medievalists tout court. It provides inspiration for those who deal with political and institutional history, since it allows one to investigate in practice the dynam-

[1] Throughout the article, the term “sentence” will be used with its meaning of *a punishment that a judge gives to someone who has committed a crime*.

[2] A chief magistrate of a medieval Italian town.

[3] A historical building of Milan, from whose balcony sentences and edicts were proclaimed by the Milanese judges.

[4] *In the name of the Lord, amen.*

ics of the exercise and management of power, the men, the methods and timing through which justice is administered, including through the selection of judges (Pagnoni, 2021); at the same time, a collection of sentences issued by a city lord provides very useful elements for the study of society and economy, through the analysis and reconstruction of the type of crime, its scene and circumstances, the weapons used, the profiles of the people involved, including their reputation, qualification and profession.

In this article, we present the LiMe dataset, an annotated Latin corpus consisting of 325 judicial documents from the first volume of the *Libri sententiarum potestatis Mediolani*. We illustrate the process undertaken for digitizing the documents and annotating them with detailed information, such as entities and relations, in order to make the manuscript more accessible and valuable to researchers. The paper is structured as follows: Section 2 provides the motivations behind this research; Section 3 outlines relative contributions in the field literature; in Section 4 we define how the data has been extracted and the final structure of the LiMe dataset; Section 5 gives examples of possible statistical and machine learning applications; in Section 6 we discuss the results and the future steps.

2. Motivation

The study of society through the filter of the judicial machine allows a better understanding of the objectives of "political discipline" and the effectiveness of this governing instruments (Campisi, 2019; Luca, 2021). At the same time, the registers of sentences still preserved in the archives of Italian cities of the last centuries of the Middle Ages, constitute a valuable field of research for those who deal with the history of gender in the medieval age (Del Bo, 2021; Dean, 2008). The analysis of such documentation on the basis of the interpretative categories typical of this historiography benefits from the possibility of questioning the source on the characteristics of alleged victims and perpetrators, the type of condemnation/absolution, the granting of pardon (*gratia*), the timing of the execution of the sentence, the type of crime, the weapons used, the place and circumstances of the offence (*delictum*), single or group action, the presence of accomplices or leaders and their gender, the personal/familial condition, the words used to identify and define each person, to mention only a few aspects of the research. Starting from the identification modalities of women and men from the language of sentences, exploiting qualifying attributes, the source offers the possibility of dismantling stereotypes and historiographical clichés.

Despite their undoubted relevance, the *Libri sententiarum potestatis Mediolani* have received little, if any, historiographical attention overall. In fact, they have not been taken into account in wide-ranging studies dedicated to the subject of the documentation issued by medieval Italian judicial bodies (Giorgi et al., 2012; Lett, 2021; Dean, 2007; Vallerani, 2012) and, until very recent years, few scholars have dealt with them specifically (Verga, 1901; Santoro, 1968; Padoa-Schioppa, 1996; Covini, 2012). The first manuscript in the series contains 126 criminal sentences pronounced by the *podestà* of Milan Carlo Zen (1385). This manuscript was recently edited by (Pizzi, 2021) and analysed in (Bassani et al., 2021).

3. Related Work

Despite being a dead language with far less resources with respect to modern languages, Latin has recently received significant attention from the research community, in both the production of annotated datasets and the training of language-specific models.

3.1. Latin Corpora

Several projects are currently dealing with the digitization and annotation of a considerable amount of Latin texts, often coming from different sources, with the purpose of being explored and exploited by history and linguistics scholars. Some of these corpora mainly present detailed syntactic and morphological annotations. It is the case of the five Latin Universal Dependencies[5] treebanks: PROIEL (Haug and Jøhndal, 2008), Perseus (Bamman and Crane, 2011), ITTB (Passarotti, 2019), LLCT (Cecchini et al., 2020), UDante (Flavio et al., 2020). LatinISE (McGillivray and Kilgarriff, 2013) is a Latin corpus for Sketch Engine, gathering documents from different websites; the corpus can be searched through the usage of tokens (13 million those present in the documents), or filtered on metadata, such as the author or the time period of each work. The LIRE (Kaše et al., 2021) dataset is another example of data integration, collecting Latin inscriptions dating back to the Roman Empire from two sources: the Epigraphic Database Heidelberg[6] (EDH) and the Epigraphik Datenbank Clauss-Slaby[7] (EDCS). The Opera Latina corpus (Denooz, 2007), created and maintained by the Laboratoire d'Analyse Statistique des Langues Anciennes (LASLA) includes 154 works from 19 classical Latin authors. The recent LiLa[8] (Passarotti et al., 2020) (Linking Latin) project has the object of building

[5] https://universaldependencies.org/la/

[6] https://edh.ub.uni-heidelberg.de

[7] http://www.manfredclauss.de

[8] https://lila-erc.eu

a common knowledge base, capable of describing several scattered Latin datasets with a unique vocabulary.

There are just a few cases of Latin corpora presenting detailed annotations for a specific task. The dataset presented in (Besnier and Mattingly, 2021) contains proper nouns of people and places in three Medieval languages, Latin included; the dataset can be employed to build named entity recognition (NER) models for low-resource languages. Addressing the task of authorship analysis, MedLatinEpi and MedLatinLit (Corbara et al., 2022) are two datasets consisting of 294 and 30 curated texts, respectively, labelled with the respective author; MedLatinEpi texts are of epistolary nature, while MedLatinLit texts consist of literary comments and treatises about various subjects.

Regarding legal texts, the Justinian's Digest has been digitized and included in a relational database (Ribary, 2020): the texts can be accessed and filtered, querying information about jurists, thematic sections and compositional structure.

3.2. Latin Language Models

In recent years, both non-contextual and contextual embedding models have been exploited for the representation of Latin text. In (Burns et al., 2021) the authors train a word2vec model on a large Latin corpus, achieving state-of-art performances on synonym detection and inter-textual search. Latin BERT (Bamman and Burns, 2020) is a contextual language model for Latin, trained on a large corpus spanning over twenty-two centuries; a fine-tuned version of Latin BERT (Lendvai and Wick, 2022) has been proposed for a word sense disambiguation task.

LatinCy (Burns, 2023) is an entire Latin NLP pipeline built for the Python library spaCy (Honnibal et al., 2020): it consists of several models, capable of performing part-of-speech tagging, dependency parsing, and named entity recognition. Stanza (Qi et al., 2020) is a collection of tools and models for the linguistic analysis of many human languages, including Latin, trained on Universal Dipendencies treebanks. UDPipe (Straka, 2018) is a pipeline for tokenization, tagging, lemmatization and dependency parsing, trainable on CoNLL-U files.

Shared tasks are being proposed in order to foster research in the field of language technologies for Classical languages. The EvaLatin 2022 Evaluation Campaign (Sprugnoli et al., 2022) proposed three tasks relative to lemmatization, part-of-speech tagging, and features identification.

4. Dataset

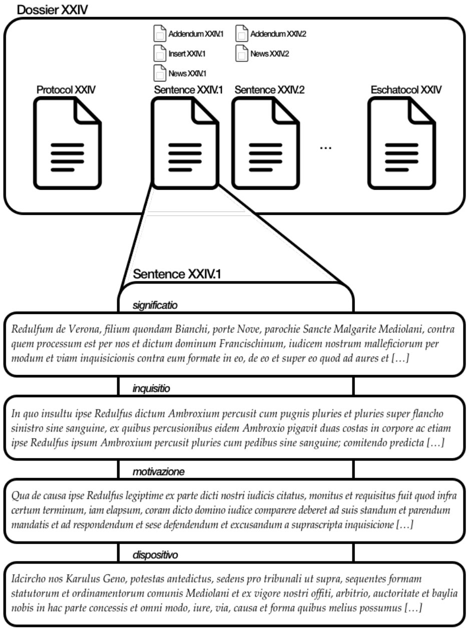
LiMe[9] (Bassani et al., 2024) is a publicly available Latin corpus consisting not only of criminal sentences, but also of many additional notes gathered from the first manuscript of the *Liber sententiarum potestatis Mediolani* (1385-1429), the oldest known registers of criminal sentences for the city of Milan. The original source, preserved in very good conditions and presenting just three mutilated texts, has been edited and transcribed in the curated edition (Pizzi, 2021). The texts have then been digitized and annotated in the context of the Fight Against Injustice Through Humanities (FAITH) project (Ferrara et al., 2023b), whose main objective is to provide common tools and methodology for the collection, digitization and integration of different historical sources. For each document, named entities, relations between them and events have been manually identified; moreover, the texts have been classified depending on the type of document and, in case of criminal sentences, they have been segmented according to a predefined annotation schema. The result is a collection of 325 documents, made of 87110 tokens, in Latin language. The annotations, performed by a team of experts, have been organized according to a custom schema; an example of the annotations is provided in Section 4.2.

4.1. Data Extraction

The main source of information in the manuscript are the criminal sentences, gathered in dossiers and ordered according to an arbitrary number given from the curator, e.g. *Sentenza I.1* refers to the first (1) judgment from the first (I) dossier. Each dossier is usually opened by a "protocol", i.e., a textual section in which the notary explicitly declares his identity and announces, following a very precise formulary, the name of the judge and *podestà* who presided over the trials. The "eschatocol" is the section closing each dossier, where the notary refers to the group of judgments he has transcribed, citing the witnesses present. Additionally, there are three other types of sources, constituting supplementary information to the judgements: an "addendum" is a document added later to the text of the judgment, indicating further developments happened after the end of the trial; an "insert" is a piece of text, reported within a judgment or addendum, usually certifying orders received from the *podestà*; finally, a "news" is an indirect evidence of an order or document that existed at the time but was not transcribed, useful in justifying decisions made by authority or actions taken by officials.

[9] https://doi.org/10.13130/RD_UNIMI/EN2TFH

The texts of criminal sentences, being them legal texts (thus with a rigid structure and a content pattern based on formulas), present the same sections and reflect a precise and largely stable structure. At the beginning, sometimes there it lies the *significatio*, i.e., the communication of the misdemeanor(s) to the *podestà* by a faithworthy person, the elder of the parish, in charge of the surveillance of a living area; this communication, however, did not always occur, so it is not always found in the text. The following part of the judgment, the *inquisitio*, narrates the events that occurred as they were reconstructed: here, the details regarding each misdemeanor (*misfatto*) are reported, such as the criminal offences, the perpetrator of the violence, the victim and any item involved. The motivational section (*motivazioni*), usually introduced by the words *qua de causa* ("the cause of"), *et predicta* ("and the aforesaid") or *et constat nobis* ("and it is agreed with us"), states the reason why the verdict was reached. Finally, the last part of the sentence consists of the decision (*dispositivo*) of conviction or acquittal and, in the former case, also of the type and amount of punishment; it generally begins with the word *idcirco* ("therefore", "about that"). A summary of the structure of a typical dossier with details on the form of a judgment is depicted in Figure 1.



Redulfum de Verona, filium quondam Bianchi, porte Nove, parochie Sancte Malgarite Mediolani, contra quem processum est per nos et dictum dominum Francischinum, iudicem nostrum malleficiorum per modum et viam inquisicionis contra eum formate in eo, de eo et super eo quod ad aures et […]

In quo insultu ipse Redulfus dictum Ambroxium percusit cum pugnis pluries et pluries super flancho sinistro sine sanguine, ex quibus percusionibus eidem Ambroxio pigavit duas costas in corpore ac etiam ipse Redulfus ipsum Ambroxium percusit pluries cum pedibus sine sanguine; comitendo predicta […]

Qua de causa ipse Redulfus legiptime ex parte dicti nostri iudicis citatus, monitus et requisitus fuit quod infra certum terminum, iam elapsum, coram dicto domino iudice comparere deberet ad suis standum et parendum mandatis et ad respondendum et sese defendendum et excusandum a suprascripta inquisicione […]

Idcircho nos Karulus Geno, potestas antedictus, sedens pro tribunali ut supra, sequentes formam statutorum et ordinamentorum comunis Mediolani et ex vigore nostri offiti, arbitrio, auctoritate et baylia nobis in hac parte concessis et omni modo, iure, via, causa et forma quibus melius possumus […]

Figure 1: Example of the structure of a dossier and details on judgment's sections.

The text of each source, strictly written in Latin language, has been thoroughly studied by experts, combining the findings extracted from the text with their domain knowledge in order to provide accurate and detailed annotations about people, places and items. For each person involved in the facts, demographic and social information have been identified: name and nicknames, biological gender, social class (*dominus*), profession, place of origin or residency, possible relationships with relatives, and roles played in the events. For instance, we know that *Laurentius de Roncho*, also referred to as *Beleius* and son of *Belollus*, was murdered in March 1385 by *Iohanollus de Raude*, also known as *Barachinus*.

Knowledge about places is important to understand where crimes were being committed and the geographical origin of the criminals: places inside the city regard the *parochiae* (parishes) and *portae* (gates), that were used to divide the territory of Milan; places outside the city are used for both towns under the jurisdiction of Milan, and for cities inside or outside of Italy; finally, generic places are used to indicate where a misdemeanor has taken place, e.g., a public street or a private house. The murderer of *Laurentius de Roncho* took place in a public street near its residency, in *Parochia Sancti Babile foris*, *Porta Horientalis*.

Within the narrative of a criminal event, it is possible to read about items used within an assault or that had been stolen by pickpockets, along with the indication of the body parts struck or striking. Additionally, for stolen artifacts, it is also specified their value, expressed in the currency of the time. For example, *Iohanollus de Raude* struck *Laurentius de Roncho* dead in the occipital bone (*in capite de retro*) with a tuck (*stocho*), an ancient type of longsword.

4.2. Annotation Structure

The annotation activity has been performed by a team of domain experts, that defined and mutually agreed on the custom guidelines followed throughout the entire process. The resulting dataset consists of a collection of 325 documents, of which most comprise the Latin text, the document type, named entities, events, relations, and text segmentation labels.

The documents are classified according to the six document types identified at the beginning of the previous section; the counts of documents for each type is reported in Table 1.

Type	Count
Sentences	127
Addendum	71
News	48
Protocol	30
Insert	26
Eschatocol	23

Table 1: The list of document types ordered by number of occurrences.

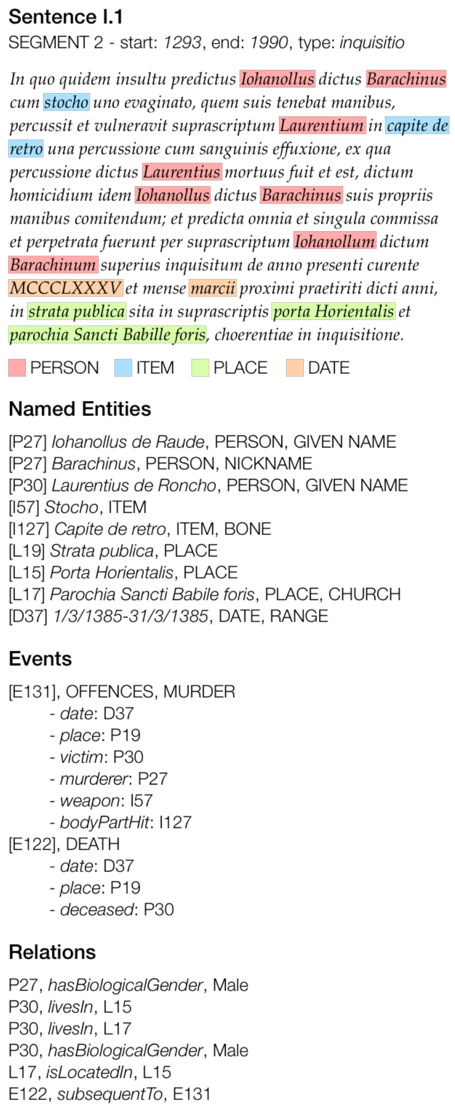
Objects under the “news” type, given the fact that they are orders or information from non transcribed documents, do not have any text; thus, knowledge about “news” can be indirectly acquired from the text of another object they refer to, usually an “addendum”. However, this knowledge is still reported in the “news” object in order to keep it logically distinct from the others.

In each document, there are eight types of named entity recognised: “PERSON” (e.g. *Laurentius de Roncho*), “PLACE” (*Parochia Sancti Babile foris*), “DATE” (*01/03/1385-31/03/1385*), “ITEM” (*stocho*), “ANIMAL” (*equum brunum*, brown horse), “MEASURE” (*valoris*, value), “UNITY OF MEASURE” (*librarum imperialum*, imperial pounds), “QUANTITY” (*viginti quinque*, twenty-five). For some of them, further sub-types have been defined, such as “GIVEN NAME” and “NICKNAME” for “PERSON”, or “CITY” and “CHURCH” for “PLACE”. The counts of named entities types and subtypes is reported in Table 2; since the same named entity can occur in multiple documents, the counts refer to the unique occurrences in the entire dataset.

Type	Sub-Type	Count
PERSON	GIVEN NAME	721
	NICKNAME	30
	NAME VARIANT	7
PLACE	CHURCH	75
	GENERIC	37
	CITY	18
DATE	DAY	105
	RANGE	42
ITEM	GENERIC	45
	BONE	25
QUANTITY	GENERIC	38
UNIT OF MEASURE	GENERIC	10
MEASURE	GENERIC	7
ANIMAL	GENERIC	3

Table 2: The list of named entity types and subtypes ordered by the number of unique occurrences.

Events are the most complex structure in the dataset; each of them is characterised of a type, usually of a subtype, and one or more arguments. There are 5 types of events: “TRIAL STAGE”, “TRIAL INTEGRATION”, “ESCHATOCOL”, “OFFENCES”, and “DEATH”. A type of event may have one or multiple subtypes, for a total of 37 event subtypes: for example, an event of type “OFFENCES” may be, among others, of subtype “INSULT”, “MURDER” or “THEFT”. Depending on its type and subtype, an event has a different set of attributes, each of them having a role and an entity playing that role: in a “THEFT” event, we expect to have a time and place of the event, a victim, a thief, and the object or quantity of money stolen.



Sentence I.1

SEGMENT 2 - start: *1293*, end: *1990*, type: *inquisitio*

> *In quo quidem insultu predictus Iohanollus dictus Barachinus cum stocho uno evaginato, quem suis tenebat manibus, percussit et vulneravit suprascriptum Laurentium in capite de retro una percussione cum sanguinis effuxione, ex qua percussione dictus Laurentius mortuus fuit et est, dictum homicidium idem Iohanollus dictus Barachinus suis propriis manibus comitendum; et predicta omnia et singula commissa et perpetrata fuerunt per suprascriptum Iohanollum dictum Barachinum superius inquisitum de anno presenti curente MCCCLXXXV et mense marcii proximi praetiriti dicti anni, in strata publica sita in suprascriptis porta Horientalis et parochia Sancti Babille foris, choerentiae in inquisitione.*

🟥 PERSON 🟦 ITEM 🟩 PLACE 🟧 DATE

Named Entities

[P27] *Iohanollus de Raude*, PERSON, GIVEN NAME
[P27] *Barachinus*, PERSON, NICKNAME
[P30] *Laurentius de Roncho*, PERSON, GIVEN NAME
[I57] *Stocho*, ITEM
[I127] *Capite de retro*, ITEM, BONE
[L19] *Strata publica*, PLACE
[L15] *Porta Horientalis*, PLACE
[L17] *Parochia Sancti Babile foris*, PLACE, CHURCH
[D37] *1/3/1385-31/3/1385*, DATE, RANGE

Events

[E131], OFFENCES, MURDER
- *date*: D37
- *place*: P19
- *victim*: P30
- *murderer*: P27
- *weapon*: I57
- *bodyPartHit*: I127

[E122], DEATH
- *date*: D37
- *place*: P19
- *deceased*: P30

Relations

P27, *hasBiologicalGender*, Male
P30, *livesIn*, L15
P30, *livesIn*, L17
P30, *hasBiologicalGender*, Male
L17, *isLocatedIn*, L15
E122, *subsequentTo*, E131

Figure 2: Example of the annotations of a segment taken from *Sentence I.1*.

Relations between entities are defined by a triple of the form (“ENTITY1”, “PREDICATE”, “ENTITY2”), where “ENTITY1” is one of the named entities or events, “PREDICATE” defines the type of relation, and “ENTITY2” can be a named entity (or event) or a group. For instance, *Laurentius de Roncho isSonOf Belollus* or *Laurentius de Roncho* *hasBiologicalGender Male*. In the dataset there are 37 unique predicates, which define 3397 unique relations.

Finally, for documents of type “sentences”, the text has been divided into segments, each of them classified with a label that specifies the section in which they appear, according to the annotation schema defined in the previous section: *significatio*, *inquisitio*, *motivazioni*, *dispositivo*. The segments are outlined by a starting and ending index, enclosing a specific span of text.

An example of all the annotations that can be found in a text is portrayed in Figure 2: this shows the amount of details that can be extracted even from a very short piece of text, like the one presented.

5. Applications

In this section we provide examples of some possible use cases for the LiMe dataset, starting from simple exploratory analysis, that can be useful for medievalist researchers, to more elaborate Natural Language Processing (NLP) tasks.

5.1. Exploratory Analysis

The detail of annotation in the LiMe dataset allow for a methodological and technical study about social, demographic, judicial and economical aspects of the city of Milan in the XIV century. Extracting all the events of type “OFFENCES”, and grouping them by subtype, it is possible to have an overview of the nature of crimes at the time. As shown in Figure 3, beside some usual types of crime, such as insults, murders and thefts, there are some kinds of particular crimes, typical of that period, such as *decapilatio*, the act of pulling someone’s hair, and *descapuzatio*, which consists in stealing a wool hat.

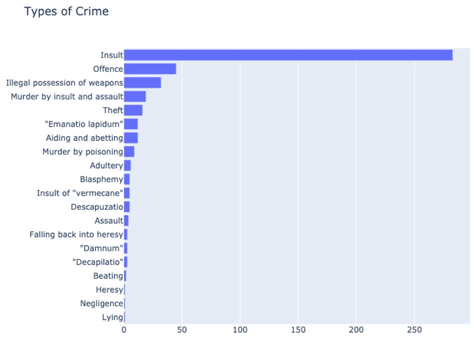


Figure 3: Number of criminal offences by type.

There are also some kinds of condemnation typical of the time, like flogging or corporal punishment (Figure 4).

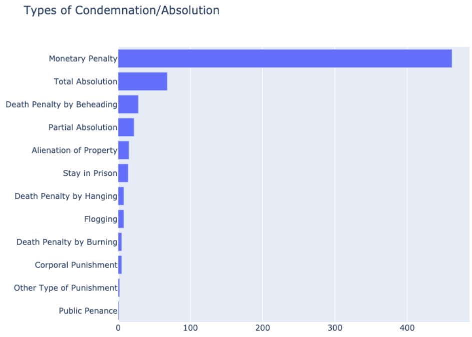


Figure 4: Number of condemnation/absolution by type.

It is also interesting to notice the difference in gender distribution of victims and criminals: despite them being mainly men in both cases, the percentage of females is almost triplicated when it comes to victims (Figure 5).

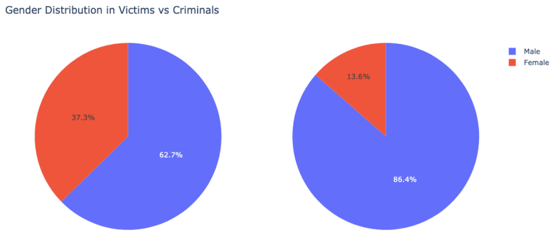


Figure 5: Distribution of males and females in victims (left) and criminals (right).

5.2. NLP Tasks

Given the peculiarity of the dataset, we believe that LiMe can be employed for many machine learning tasks involving the usage of NLP techniques. Here we provide two examples of traditional problems: document classification and text segmentation.

5.2.1. Document Classification

A document classification task regards the process of automatically assigning predefined labels to documents based on their content. For this reason, we decided to employ the 276 documents having a text, leaving out the “news” documents and ending up with five possible labels: “addendum”, “eschatocol”, “insert”, “protocol”, “sentence”. We employ Latin BERT (Bamman and Burns, 2020), a contextual language model trained on a large corpus in Latin language, and fine-tune it on the training set (221 documents) for this specific classification task. The model achieves a weighted F1 score of 0.96 on the test set (55 documents), performing remarkably well on every class (Figure 6).

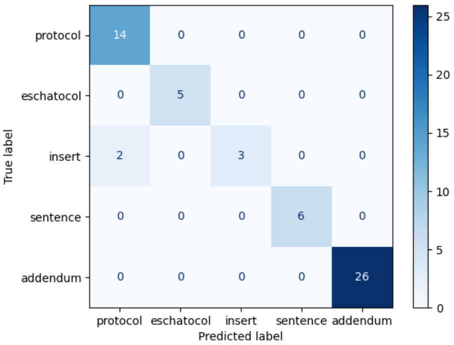


Figure 6: Confusion matrix for the document classification task.

5.2.2. Text Segmentation

A text segmentation task consists in dividing a given text into meaningful and coherent segments based on an underline annotation schema. The documents interested by this task are the “sentences” that, together, are made of more than one thousand textual segments. Each of them has a section associated to it, according to the following schema: “significatio”, “inquisitio”, “motivazioni”, “dispositivo”. In order to solve the task, we employ Rewired Conditional Random Fields (Ferrara et al., 2023a), a recent approach developed for the textual segmentation of Italian judgments, capable of working in a few-shot scenario, which is ideal given the low number of available observations. We train the above model on the segments of one hundred “sentences”: the model achieves a weighted F1 score of 0.84 on the remaining 20% of the dataset left out for evaluation purposes (Figure 7).

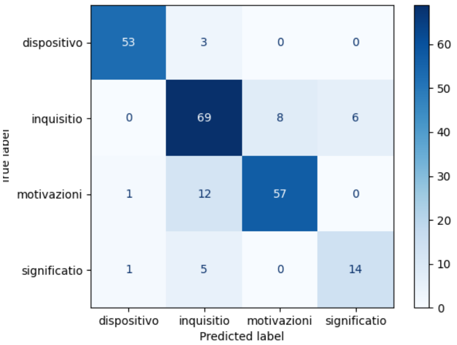


Figure 7: Confusion matrix for the text segmentation task.

6. Conclusion

The *Libri sententiarum potestatis Mediolani* are a valuable resource not only for scholars studying medieval law, but also for historians and linguists. The LiMe dataset proves how the digitisation and annotation of these kinds of sources allow for a methodological and technical analysis of the data, thanks to the usage of statistical and machine learning tools. In the future, we expect to: exploit the current dataset for more complex tasks, such as named entity recognition or event extraction; increase the number of annotated documents, with information coming from subsequent volumes of the *Libri*, which are currently being examined by experts; extend the current annotations with features at syntactical and morphological levels.

7. References

David Bamman and Patrick J. Burns. 2020. Latin BERT: A Contextual Language Model for Classical Philology. *arXiv e-prints*, page arXiv:2009.10053.

David Bamman and Gregory Crane. 2011. The ancient greek and latin dependency treebanks. In *Language Technology for Cultural Heritage*, pages 79–98, Berlin, Heidelberg. Springer Berlin Heidelberg.

Alessandra Bassani. 2021. Le assoluzioni nel Liber comunis potestatis Mediolani: riflessioni sull’ipotesi di una giustizia giusta. *Notariorum Itinera*, 7:177–204.

Alessandra Bassani, Marta Calleri, and Marta L. Mangini. 2021. Liber sententiarum potestatis mediolani (1385): Storia, diritto, diplomatica e quadri comparativi. *Notariorum Itinera*, 7.

Alessandra Bassani, Beatrice Del Bo, Alfio Ferrara, Marta Mangini, Sergio Picascia, and Ambra Stefanello. 2024. LiMe - Liber sententiarum potestatis Mediolani.

Clément Besnier and William Mattingly. 2021. Named-entity dataset for medieval latin, middle high german and old norse. *Journal of Open Humanities Data*, 7(0):23.

Raffaella Bianchi Riva. 2021. Iniuria e insultus tra diritto e politica. Le offese alle magistrature comunali nella legislazione statutaria e nella prassi giudiziaria in età viscontea. *Notariorum Itinera*, 7:239–264.

Patrick J. Burns. 2023. Latincy: Synthetic trained pipelines for latin nlp.

Patrick J. Burns, James A. Brofos, Kyle Li, Pramit Chaudhuri, and Joseph P. Dexter. 2021. Profiling of intertextuality in latin literature using word embeddings. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4900–4907.

Luca Campisi. 2019. Prassi giudiziaria a vercelli nel xiv secolo. *Studi di storia medioevale e di diplomatica - Nuova Serie*, (2):131–150.

Flavio Massimiliano Cecchini, Timo Korkiakangas, and Marco Passarotti. 2020. A new Latin treebank for Universal Dependencies: Charters between Ancient Latin and Romance languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 933–942, Marseille, France. European Language Resources Association.

Silvia Corbara, Alejandro Moreo, Fabrizio Sebastiani, and Mirko Tavoni. 2022. Medlatinepi and medlatinlit: Two datasets for the computational authorship analysis of medieval latin texts. *Journal on Computing and Cultural Heritage*, 15(3):1–15.

Nadia Covini. 2012. Assenza o abbondanza? la documentazione giudiziaria lombarda nei fondi notarili e nelle carte ducali (stato di milano, xiv-xv secolo). *La documentazione degli organi giudiziari*, pages 483–499.

Trevor Dean. 2007. *Crime and Justice in Late Medieval Italy*. Cambridge University Press.

Trevor Dean. 2008. Theft and gender in late medieval bologna. *Gender & History*, 20(2):399–415.

Beatrice Del Bo. 2021. Tutte le donne (del registro) del podestà fra cliché e novità. *Notariorum Itinera*, 7:83–106.

Joseph Denooz. 2007. Opera latina: le nouveau site internet du lasla. *Journal of Latin Linguistics*, 9(3):21–34.

Alfio Ferrara, Sergio Picascia, and Davide Riva. 2023a. Few-shot legal text segmentation via rewiring conditional random fields: A preliminary study. In *Advances in Conceptual Modeling*, pages 141–150, Cham. Springer Nature Switzerland.

Alfio Ferrara, Sergio Picascia, Elisabetta Rocchetti, Gaia Varese, et al. 2023b. The FAITH project: integrated tools and methodologies for digital humanities. *Proceedings of the Statistics and Data Science Conference*, pages 323–327.

Cecchini Flavio, Rachele Sprugnoli, Moretti Giovanni, Passarotti Marco, et al. 2020. Udante: First steps towards the universal dependencies treebank of dante's latin works. In *Proceedings of the Seventh Italian Conference on Computational Linguistics*, pages 99–105. Accademia University Press.

Andrea Gamberini. 2014. *A Companion to Late Medieval and Early Modern Milan: The Distinctive Features of an Italian State*, volume 7. Brill.

Andrea Giorgi, Stefano Moscadelli, and Carla Zarrilli. 2012. *La documentazione degli organi giudiziari nell'Italia tardo-medievale e moderna: atti del convegno di studi, Siena, Archivio di Stato, 15-17 settembre 2008*. Number v. 1 in Pubblicazioni degli Archivi di Stato. Saggi. Ministerio per i beni e le attività culturali, direzione generale per gli archivi.

Dag TT Haug and Marius Jøhndal. 2008. Creating a parallel treebank of the old indo-european bible translations. In *Proceedings of the second workshop on language technology for cultural heritage data (LaTeCH 2008)*, pages 27–34.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Roberto Isotton. 2021. La repressione dei reati di furto e rapina nel Liber sententiarum potestatis Mediolani del 1385: acquisizioni e questioni aperte. *Notariorum Itinera*, 7:205–238.

Vojtěch Kaše, Petra Heřmánková, and Adéla Sobotková. 2021. Classifying latin inscriptions of the roman empire: A machine-learning approach. In *Proceedings of the Conference on Computational Humanities Research 2021CEUR-WS*, volume 2989, pages 123–135.

Piroska Lendvai and Claudia Wick. 2022. Finetuning Latin BERT for word sense disambiguation on the thesaurus linguae latinae. In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, pages 37–41, Taipei, Taiwan. Association for Computational Linguistics.

Didier Lett. 2021. *I Registri Della Giustizia Penale Nell'Italia Dei Secoli XII-XV*. Ecole française de Rome.

Campisi Luca. 2021. *L'impatto sociale. I protagonisti delle pratiche giudiziarie a Vercelli fra XIV e XV secolo*. Phd thesis, Università degli Studi di Milano.

Barbara McGillivray and Adam Kilgarriff. 2013. Tools for historical corpus research, and a corpus of latin. *New methods in historical corpus linguistics*, 1(3):247–257.

Giovanni Minnucci. 2021. Intorno al Liber sententiarum potestatis Mediolani e ad altre fonti giudiziarie. Alcune note conclusive. *Notariorum Itinera*, 7:373–380.

Antonio Padoa-Schioppa. 1996. *La giustizia milanese nella prima età viscontea (1277-1300)*. Giuffrè.

Antonio Padoa-Schioppa. 2017. *A History of Law in Europe: From the Early Middle Ages to the Twentieth Century*. Cambridge University Press.

Fabrizio Pagnoni. 2021. Selezione e circolazione dei giudici ai malefici nel dominio visconteo fra Tre e Quattrocento. *Notariorum Itinera*, 7:61–81.

Marco Passarotti. 2019. *The Project of the Index Thomisticus Treebank*, pages 299–320. De Gruyter Saur, Berlin, Boston.

Marco Passarotti, Francesco Mambrini, Greta Franzini, Flavio Massimiliano Cecchini, Eleonora Litta, Giovanni Moretti, Paolo Ruffolo, and Rachele Sprugnoli. 2020. Interlinking through lemmas. the lexical collection of the lila knowledge base of linguistic resources for latin. *Studi e Saggi Linguistici*, 58(1):177–212.

Pier Francesco Pizzi. 2021. *Liber sententiarum potestatis Mediolani (1385), Edizione critica*. Società Ligure di Storia Patria.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Marton Ribary. 2020. A relational database of roman law based on justinian's digest. *Journal of Open Humanities Data*, 6(1):5.

C. Santoro. 1968. *Gli offici del comune di Milano e del dominio visconteo sforzesco (1216-1515)*. 1. collana: Monografie, ricerche ausilierie, opere strumentali. A. Guiffrè.

Rachele Sprugnoli, Marco Passarotti, Flavio Massimiliano Cecchini, Margherita Fantoli, and Giovanni Moretti. 2022. Overview of the EvaLatin 2022 evaluation campaign. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 183–188, Marseille, France. European Language Resources Association.

Claudia Storti. 2021. 1385: un anno tra politica e giustizia a milano. *Notariorum Itinera*, 7:7–31.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

M. Vallerani. 2012. *Medieval Public Justice*. Studies in Medieval & Early Mo. Catholic University of America Press.

Chiara Valsecchi. 2021. «Per viam inquisicionis». Note sul processo criminale a Milano in un'età di transizione. *Notariorum Itinera*, 7:127–176.

E. Verga. 1901. *Le sentenze criminali dei podestà milanesi 1385-1429: appunti per la storia della giustizia punitiva in Milano*. P. Confalonieri.

The Rise and Fall of Dependency Parsing in Dante Alighieri's *Divine Comedy*

Claudia Corbetta[1], Marco Passarotti[2], Giovanni Moretti[2]
Università di Pavia/Bergamo[1], Università Cattolica del Sacro Cuore[2],
claudia.corbetta@unibg.it {marco.passarotti, giovanni.moretti}@unicatt.it

Abstract

In this paper, we conduct parsing experiments on Dante Alighieri's *Divine Comedy*, an Old Italian poem composed between 1306-1321 and organized into three *Cantiche —Inferno*, *Purgatorio*, and *Paradiso*. We perform parsing on subsets of the poem using both a Modern Italian training set and sections of the *Divine Comedy* itself to evaluate under which scenarios parsers achieve higher scores. We find that employing in-domain training data supports better results, leading to an increase of approximately +17% in Unlabeled Attachment Score (UAS) and +25-30% in Labeled Attachment Score (LAS). Subsequently, we provide brief commentary on the differences in scores achieved among subsections of *Cantiche*, and we conduct experimental parsing on a text from the same period and style as the *Divine Comedy*.

Keywords: Parsing, Dependency syntax, Old Italian, Modern Italian, *Divine Comedy*

1. Introduction

The *Divine Comedy*[1], an Old Italian[2] poem authored by Dante Alighieri, was composed in the period between 1306 and 1321. This seminal work comprises three *Cantiche*: *Inferno*, *Purgatorio*, and *Paradiso*. Each *Cantica* is subdivided into *Canti*, culminating in a total of 100 (34 in *Inferno*, 33 in *Purgatorio*, and 33 in *Paradiso*)[3]. Recognized as a foundational pillar of Italian literature, the language of the *Divine Comedy* plays a pivotal role in the evolution of the Italian language.

A linguistic annotation of the *Divine Comedy* is provided by DanteSearch (Tavoni, 2011), an online corpus[4] containing all the works of Dante Alighieri. DanteSearch employs a tagset to identify parts of speech (PoS) and morphological features of words[5] and provides a clause-based syntactic annotation style, wherein the functions of clauses within the sentence (e.g., declarative, interrogative, exclamative) are recorded[6].

Nevertheless, the annotation schema and tagset of DanteSearch are not fully compatible with other styles, such as the one used in the Universal Dependencies initiative[7] (UD), which is currently the standard de facto schema for syntactically annotated corpora. UD is an annotation framework designed to establish a universal formalism for dependency-based syntactic annotation (De Marneffe et al., 2021). Its primary objective is to facilitate cross-linguistic comparison, starting by collecting linguistic information into a treebank, a linguistically annotated corpus containing several layers of annotation such as lemmatization, PoS and (dependency) syntax annotation.

In the UD collection, the first and sole treebank documenting Old Italian is the *Divine Comedy*. Specifically, this treebank, referred to as Italian-Old in UD, encompasses the first *Cantica* of the *Divine Comedy*, namely *Inferno*. The creation of the treebank for the *Divine Comedy* (Corbetta et al., 2023) leveraged pre-existing annotated data from DanteSearch. While PoS and lemmas were semi-automatically converted from DanteSearch to the UD format, the dependency-based syntactic annotation was conducted anew.

Besides the need to change the syntactic annotation style from clause level to word level[8], the UD-like annotation of *Inferno* was conducted fully

[1] This paper is the result of the collaboration between the three authors. For academic purposes, Claudia Corbetta is responsible of Sections 2,3,4; Marco Passarotti of Sections 1,5; Giovanni Moretti developed the tri-gram and sub-tree extraction script and built the Stanza Model of *Inferno* IV-XXXIV. Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[2] In this paper, the language of the *Divine Comedy* is referred to as Old Italian. For an in-depth understanding of the language of the *Divine Comedy*, see (Manni, 2013).

[3] Refer to (Inglese, 2012) for an introductory overview of the poem.

[4] `https://dantesearch.dantenetwork.it`

[5] To gain a deeper understanding of the concept of "word" as attested in DanteSearch, we refer to (Tavoni, 2011).

[6] For a comprehensive understanding of the clause-based annotation scheme, please see (Gigli, 2004).

[7] `https://universaldependencies.org.`

[8] As previously mentioned, the clause-based syntactic annotation style utilized by DanteSearch is not compatible with that of UD, which is word-based. For a more in-depth understanding of the distinction, please see (Corbetta et al., 2023).

manually for two main reasons: (i) to enhance the annotators' skills through steady confrontation with data; and (ii) to prevent biases in the annotation work that could arise from using a pre-parsed text. Specifically, we did not use the trained models of parsers developed from the UD treebanks for Modern Italian.

Having completed the manual annotation for *Inferno*, this paper evaluates the performance of models trained on Modern Italian treebanks available in UD, as well as models trained on subsets of *Inferno* itself. This evaluation aims to ascertain whether one approach is preferred over the other for assisting in the annotation of the remaining parts of the *Divine Comedy*, specifically *Paradiso*[9]. Additionally, in the context of future work, we aim to explore whether using a parser based on the *Divine Comedy* could be beneficial for annotating similar texts from the same period.

The paper is organized as follows. Section 2 describes tests of parsing on *Inferno* with Modern Italian data. Section 3 describes how we selected the subset upon which we conduct parser experiments and illustrates how we calculated the correlation degree among the subset and their respective *Cantica*. Section 4 reports the results of experimenting parsing respectively with models trained on the *Divine Comedy* data and with models trained on Modern Italian data. We compare scores among the *Cantiche* and conduct parsing tests on a poem from the same period. The final section 5 summarizes the results and highlights future directions of research.

2. Parsing *Divine Comedy* Text with Modern Italian Data: a Journey through *Inferno*

The comparison between Old Italian and Modern Italian, particularly concerning syntax, has been a topic of debate[10]. The examination of potential distinctions between Old Italian and Modern Italian language lies beyond the scope of this paper. Our current investigation focuses on evaluating the syntactic accuracy of models trained on Modern Italian data for parsing *Inferno*.

While in UD the sole treebank containing Old Italian data is Italian-Old, consisting of *Inferno*, Modern Italian is covered by multiple treebanks, representing diverse styles and genres[11]. We specify that among all Modern Italian treebanks, none represents the same genre as *Divine Comedy*, namely the poetic genre, which might affect negatively the accuracy rates of the trained models.

As *Inferno* is the sole manually annotated treebank of Old Italian available, we test the accuracy of parsers using a training set based on Modern Italian data. We parse *Inferno* using two different parsers. We employ UDPipe1 (Straka et al., 2016; Straka and Straková, 2017), which is a trainable pipeline for tokenization, tagging, lemmatization and dependency parsing of CoNLLU-files[12], and Stanza (Qi et al., 2020), a neural network pipeline, that includes, among other functionalities, tokenization, tagging, lemmatization and dependency parsing[13]. For both UDPipe1 and Stanza, we only perform parsing, retaining the tokenization, lemmas, PoS and morphological features of the manually annotated text. We build models using UDPipe1 and Stanza based on training sets provided by three major Modern Italian treebanks (six models in total): ISDT (Bosco et al., 2013), VIT (Tonelli et al., 2008) and Par-TUT(Bosco et al., 2012)[14]. We evaluate the performance of the two parsers, by averaging the accuracy rates of their trained models (two evaluation rates in total). To evaluate the accuracy of the output, we rely on `eval.py`[15].

Table 1 reports the scores.

Inferno	UDPipe1	Stanza
UAS	65.28	65.16
LAS	56.98	50.85

Table 1: Accuracy metrics of *Inferno* with UDPipe1 and Stanza.

Considering that the average UAS (Unlabeled Attachment Scores) and LAS (Labeled Attachment

[9] We completed the annotation of *Purgatorio* and it is scheduled for publication in the upcoming next release of UD. See https://universaldependencies.org/release_checklist.html.

[10] We refer to the Preface of (Salvi and Renzi, 2010) for an introduction to Old Italian and its differences with Modern Italian. For an overview of syntactic peculiarities of Old Italian syntax, we refer to (Dardano and Frenguelli, 2002).

[11] For detailed information about Modern Italian treebanks in UD, see https://universaldependencies.org/it/index.html.

[12] https://github.com/ufal/udpipe.

[13] https://stanfordnlp.github.io/stanza/index.html.

[14] Refer to https://github.com/UniversalDependencies/UD_Italian-ISDT for ISDT; https://github.com/UniversalDependencies/UD_Italian-VIT for VIT and https://github.com/UniversalDependencies/UD_Italian-parTUT for Par-TUT.

[15] The `eval.py` is designed to assess the accuracy of a UD tokenizer, lemmatizer, tagger and parser against a gold-standard data. The script is available at https://github.com/UniversalDependencies/tools/blob/master/eval.py.

Scores)[16] are around 65.22 and 53.91 respectively, we have decided to challenge the results for Modern Italian by attempting to increase the scores. To do so, we utilize samples from the *Divine Comedy* as training set.

3. Data: Evaluating Correlation Degree

We select a subset of three *Canti* as test set, comprising 9% of the respective *Cantiche*[17], which we demonstrate to be adequately representative of their respective *Cantica*.

In order to evaluate the correlation degree of each subsection with its respective *Cantica*, we examine tri-gram variation in PoS tagging and subtree label attachment. The evaluation is performed by calculating the Pearson correlation coefficient for each measure, comparing the linguistic features of the subset with those of its corresponding *Cantica*. This approach provides a quantitative measure of how closely the linguistic characteristics align between the subsection and the complete *Cantica*.

3.1. Part of Speech Tri-gram Detection

We assess the degree of correlation for tri-gram PoS by converting DanteSearch tagset into UD PoS. For this task, a direct automated conversion from DanteSearch to UD PoS is applied. This means that the conversion was performed without considering the different criteria of PoS assignment between DanteSearch and UD. For instance, we do not differentiate cases such as possessive adjectives, which are tagged as adjectives in DanteSearch but classified as determiners in UD[18].

More specifically, the tri-grams analysis of PoS is conducted on the subset of *Canti* I-III of *Inferno*, *Purgatorio* and *Paradiso*, corresponding to the aforementioned 9% of the *Cantiche*. This analysis is then compared with the tri-gram distribution of the respective *Cantica*.

PoS tri-grams are extracted at sentence level, using full stops for sentence splitting[19]. For instance, Table 2 reports PoS tri-grams for the following sentence.

> *Se' savio; intendi me' ch'i' non ragiono.*
> (*Inf*. II, v. 36)
>
> You're wise; you know far more than what
> I say.

Tri-gram of words	PoS tri-gram
Se'/savio/intendi	AUX/ADJ/VERB
savio/intendi/me'	ADJ/VERB/ADV
intendi/me'/ch'	VERB/ADV/SCONJ
me'/ch'/i'	ADV/SCONJ/PRON
ch'/i'/non	SCONJ/PRON/ADV
i'/non/ragiono	PRON/ADV/VERB

Table 2: The extraction of tri-grams from a sentence in *Inferno*.

Tri-grams of each subsection are then listed according to their frequency and compared with the tri-gram rankings of the respective *Cantica*, evaluating the Pearson correlation coefficient (Brezina, 2018) to estimate their correlation degree. To mitigate data sparsity due to the different size of the texts compared, we exclude the tri-grams belonging to the less frequent 5% of the total[20].

Table 3 reports the Pearson correlation of each subset in respect with its *Cantica*.

Inferno	Purgatorio	Paradiso
0.835	0.845	0.868

Table 3: Pearson correlation for tri-gram PoS.

As Pearson coefficient is > 0.5 (Brezina, 2018, p. 144), we can consider the correlation to be strong and generalize that the PoS tri-gram distribution of each subset of *Canti* I-III correlates with its respective *Cantica*.

3.2. Sub-tree Label Attachment

We also assess the correlation degree by examining the syntactic structure, specifically sub-tree dependency relations, in the subsection I-III of *Inferno* and I-III *Purgatorio* compared with the corresponding *Cantica*. We abstain from conducting correlation for *Paradiso* since its syntax is presently under development. We assume that the sub-tree label correlation evaluated within *Inferno* and *Purgatorio* could be consistent with the other *Cantica*, in agreement with the results shown in the PoS correlation.

[16] Refer to (Buchholz and Marsi, 2006) for an insight into syntactic metrics.

[17] The number of tokens in each subset (I-III) is 3561 tokens for *Inferno*, 3622 for *Purgatorio*, and 3484 for *Paradiso*.

[18] The described procedure will not have a negative impact on the evaluation, as we maintain a unified PoS tagging system, specifically the one adopted by DanteSearch, and we consistently employ such scheme to analyze the tri-gram correlation within the first three *Canti*.

[19] This means that the PoS of the last word of a sentence is the final item of a tri-gram, while the PoS of the first word of a sentence serves as the first item of a tri-gram.

[20] This results in excluding around the 1700 tri-grams out of the total of approximately 32300. More precisely, we exclude 1726 out of 32564 for *Inferno*; 1729 out of 32428 for *Purgatorio* and 1701 out of 32027 for *Paradiso*.

When referring to sub-tree dependency relations, we denote a sub-tree composed of the PoS of a governor node (n1), the dependency relation[21] of the node n1 with its dependent (deprel, such as `nsubj` for the subject relation), and the PoS of the dependent node (n2), following the schema:

ragiono –> `nsubj` –> i'
VERB –> `nsubj` –> PRON

More precisely, the extraction of sub-tree labels is performed for each syntactic node (except for punctuation, marked with the deprel `punct`). For each node, a triple is extracted, consisting of the node (n1), a dependent node (n2) and their dependency relation. Subsequently, we derive the PoS of the involved nodes.

Following the approach used for tri-grams, we subsequently apply Pearson correlation to assess the correlation degree of sub-tree labels between the two. Similarly to the PoS tri-gram, we exclude sub-trees that belong to the least frequent 5% of the total[22]. Pearson correlation showed in Table 4 is > 0.5, namely 0.744 for *Inferno* and 0.737 for *Purgatorio*, highlighting a strong correlation between the sub-tree dependency labels of the first three *Canti* of *Inferno* and *Purgatorio* and their entire *Cantica*.

Inferno	**Purgatorio**
0.772	0.794

Table 4: Pearson correlation for sub-tree labels.

Given the high Pearson coefficient observed in both tri-gram correlation and sub-tree labels (limited to *Inferno* and *Purgatorio*), we conclude that the first three *Canti* might be partially considered representative of the respective *Cantica*.

4. Parsing and Evaluation: Examining the First Three *Canti*

Given the high correlation degree between the first three *Canti* and their respective *Cantiche*, we proceed with parsing experiments and evaluation metric checks to see whether, by using subsets of the *Divine Comedy* as training data, we can improve the UAS and LAS scores obtained with Modern Italian training data and reported in Table 1.

4.1. *Divine Comedy* on *Divine Comedy*

We train both UDPipe1 (UDP) and Stanza (Stan) on a training set consisting of *Canti* IV-XXXIV of *Inferno*, encompassing the 30% of the all *Divine Comedy* and 91% of *Inferno*[23]. Subsequently, we test the two models on the first three *Canti* of each *Cantica*, namely *Inferno* I-III (Inf), *Purgatorio* I-III (Purg) and *Paradiso* I-III (Par) and evaluate the syntax metrics, namely LAS and UAS, with respect to the gold standard of each test set[24]. The evaluation is performed using `eval.py` for the output of UDPipe1 model. In the case of Stanza, the evaluation is executed automatically after each training run[25].

Table 5 shows LAS and UAS of each subset, i.e., I-III of *Inferno* (Inf), *Purgatorio* (Purg) and *Paradiso* (Par), for both UDPipe1 (UDP) and Stanza (Stan) model.

It is noteworthy that the scores provided by both UDPipe1 and Stanza in Table 5 are significantly higher when compared with the scores obtain from model trained on Modern Italian data on *Inferno* (see Table 1). The boost of the Stanza model trained on the *Divine Comedy* data is 19.81 for UAS and 29.21 for LAS[26] compared to the Stanza model trained on Modern Italian data. Regarding UDPipe1, we observe an increase of 14.35 scores for UAS and 16.56 scores for LAS in favor of models trained on *Divine Comedy*[27].

We replicate the test using only the Stanza model with the same training set of Modern Italian used for the data in 2, testing it on the subsets of *Inferno* I-III, *Purgatorio* I-III and *Paradiso* I-III.

As shown in Table 6, scores obtained with the training set of Modern Italian on the subsets *Inf*, *Purg*, and *Par*, reflect the scores obtained for the parsing of *All Inferno*, reported in Table 1. This confirms that using part of the text as the training set yields better results than using Modern Italian data.

It is also interesting to note that the scores across the *Cantiche* flow both in Table 5 and Table 6, being higher for *Inferno*, followed by *Paradiso*, and then *Purgatorio*. We briefly comment fluctuations in Subsection 4.2.

[21] A list of dependency relations and the specific meaning of each label is documented in UD.

[22] This implies that we do not consider 1764 sub-trees out of 33387.

[23] The training set consists of 1118 sentences and 37806 syntactic words.

[24] The gold standards of *Purgatorio* I-III and *Paradiso* I-III were manually annotated by an annotator with competence in Old Italian.

[25] For detailed information on the evaluation in Stanza, please see https://stanfordnlp.github.io/stanza/training_and_evaluation.html#evaluation.

[26] We considered the average of both LAS and UAS scores for *Inf*, *Purg* and *Par* subsets in Table 5, precisely 84.97 for UAS and 80.06 for LAS.

[27] The average of LAS and UAS scores for the subsets *Inf*, *Purg*, and *Par* are respectively 73.54 and 79.63.

	Inf		Purg		Par	
Metr.	UDP	Stan	UDP	Stan	UDP	Stan
UAS	82.65	87.73	77.93	82.67	78.50	84.50
LAS	77.87	84.02	71.42	77.31	71.33	78.85

Table 5: LAS and UAS scores of each subset parsed with UDPipe1 and Stanza.

	Inf	Purg	Par
UAS	69.05	66.28	67.74
LAS	56.14	53.30	54.31

Table 6: LAS and UAS scores of each subset parsed with Stanza model trained on Modern Italian.

4.2. Comparing Metrics across *Cantiche*

By analyzing syntactic metrics across *Cantiche*, we notice that scores flow throughout the samples of *Inferno*, *Purgatorio*, and *Paradiso*. Such fluctuations are evident in both datasets parsed with a training dataset composed of a section of *Inferno* (Table 5) and the one trained with Modern Italian data (Table 6).

In this Section, we briefly comment on the data in Table 5, namely on metrics achieved from models trained on *Divine Comedy* data[28]. The metrics presented in Table 5 demonstrate an enhanced performance under an "in-domain" condition, specifically when the training and test sets pertain to the same *Cantica*, *Inferno*. When comparing the UAS and LAS scores of *Inferno* with those of *Purgatorio* and *Paradiso*, *Inferno*'s metrics show a boost of 4.14 (Stanza) and 4.44 (UDPipe1) in LAS, and of 5.94 (Stanza) and 6.50 (UDPipe1) scores in UAS[29].

Examining closely the differences among *Purgatorio* and *Paradiso*, we also observe that *Paradiso* outperforms *Purgatorio*. Specifically, for both UDPipe1 and Stanza models, *Paradiso* experiences an improvement of 0.57 and 1.83 in UAS, respectively. The LAS score boost achieved by the Stanza model supports the observed trend in UAS metrics, with *Paradiso* LAS achieving a superior score of 1.54 points compared to *Purgatorio*. Contrary to the trend, the UDPipe1 LAS score seems to exhibit a slightly better performance in *Purgatorio* than in *Paradiso*, but the difference of 0.09 in score is very low.

The data presented suggest that syntactic structures of *Paradiso* seem to be more akin to *Inferno* than *Purgatorio* is to *Inferno*, especially for the first three *Canti* of the *Cantica*. However, such a claim deserves to be substantiated through additional studies.

4.3. Experimenting outside the *Divine Comedy*: Testing Guido Cavalcanti's Poem

To verify the efficiency of the Stanza model trained on the *Divine Comedy* data, we test it on a text from the same period and style as Dante Alighieri's poem. We select a text by Guido Cavalcanti (1259-1300), a poet contemporary to Dante and belonging to the same socio-cultural milieu[30]. The selected text is "Voi che per li occhi mi passaste il core", a poem in Old Italian, specifically Old Florentine, consisting of 111 syntactic words.

We parse the poem with Stanza model trained on all Inferno and with Stanza models trained on different Modern Italian treebanks[31] and we evaluate the syntactic metrics. Tokenization, lemmatization, PoS tagging, and morphological features are provided to the model, which is solely tasked with performing syntactic tasks.

	Stan All Inf	Stan Mod It
UAS	86.49	66.37
LAS	75.68	48.65

Table 7: Metrics in Cavalcanti's poem with Stanza model trained on All *Inferno* and Stanza model trained on Modern Italian data.

As shown in Table 7, Stanza model trained on All Inferno performs better than Modern Italian one. The boost is significantly around 20.12 for UAS and 27.03 for LAS.

Despite the small sample size, the boost is promising. We will further investigate and experiment by testing on larger samples and expanding the domain to include more authors and texts of the same period to understand whether the *Divine Comedy* might be representative enough.

[28] Discussion of metrics achieved from Modern Italian data will be left for further studies.

[29] To calculate the boost of *Inferno*'s scores, we consider an average among the UAS and the LAS scores of *Purg* and *Par* scores.

[30] We refer to (Cavalcanti, 2011) for an introduction of Guido Cavalcanti and his rhymes.

[31] After parsing the poem with models trained on respectively ISDT, VIT, Par-TUT, we calculate an average of the scores of all Modern Italian models.

5. Conclusion

In this paper, we parse sections of the *Divine Comedy*, comparing the accuracy of models trained on Modern Italian data with those trained on portions of the *Divine Comedy* itself.

Firstly, our findings reveal that employing parsers trained on texts from the *Divine Comedy*, namely within their respective domain, result in higher accuracy. Such trend confirms the literature stating that having in-domain training data facilitates parsing results (Khan et al., 2013b,a), particularly when dealing with texts from the same author (Mambrini and Passarotti, 2012). We can therefore conclude that, at the current state of the art, despite having a larger amount of Modern Italian treebanks, using Modern Italian training set to parse the *Divine Comedy* does not result in better parsing outcomes.

Additionally, the data obtained from the comparison among the first three *Canti* of each *Cantica* highlight a greater proximity between the syntax of the first three *Canti* of *Paradiso* and the first ones of *Inferno*, compared to *Purgatorio*. However, even though we have demonstrated the representativeness of the first three *Canti* with the respective *Cantica*, the analyzed data do not allow us to identify a specific trend sufficient to draw conclusions about the possible proximity or distance between the syntax of the all three *Cantiche*.

Lastly, we conduct a brief experiment on a text contemporaneous with the *Divine Comedy*, illustrating the superiority of utilizing a model trained on similar chronological and textual types over models of Modern Italian.

As potential future work, we will investigate whether augmenting the training data by merging datasets from both Old and Modern Italian, notwithstanding the diversity in genre, will result in enhanced parsing accuracy. Moreover, further studies, along with additional annotated data[32], are necessary to ascertain the relationship between the results and the diversity of genres. Future research endeavors will be dedicated to delving deeper into these aspects.

6. Acknowledgments

We would like to thank the anonymous reviewer for the accurate suggestions.

7. Bibliographical References

Cristina Bosco, Simonetta Montemagni, and Maria Simi. 2013. Converting Italian treebanks: Towards an Italian Stanford dependency treebank. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 61–69, Sofia, Bulgaria. Association for Computational Linguistics.

Cristina Bosco, Manuela Sanguinetti, and Leonardo Lesmo. 2012. The parallel-TUT: a multilingual and multiformat treebank. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1932–1938, Istanbul, Turkey. European Language Resources Association (ELRA).

Vaclav Brezina. 2018. *Statistics in corpus linguistics: A practical guide*. Cambridge University Press.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City, NJ, USA. Association for Computational Linguistics.

Guido Cavalcanti. 2011. *Rime*. A cura di Giorgio, Inglese and Roberto, Rea. Carocci; Critical Edition.

Claudia Corbetta, Marco Carlo Passarotti, Flavio Massimiliano Cecchini, and Giovanni Moretti. 2023. Highway to hell. towards a universal dependencies treebank for dante alighieri's comedy. In *Proceedings of CLiC-it 2023: 9th Italian Conference on Computational Linguistics*, Venice, Italy. Associazione Italiana di Linguistica Computazionale.

Maurizio Dardano and Gianluca Frenguelli. 2002. *SintAnt. La sintassi dell'italiano antico*. ARACNE.

Marie-Catherine De Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Sara Gigli. 2004. Codifica sintattica della commedia dantesca. *PhD diss., Università di Pisa*.

Giorgio Inglese. 2012. *Dante: guida alla Divina Commedia. Nuova edizione*. Carocci, Roma, Italy.

Mohammad Khan, Markus Dickinson, and Sandra Kübler. 2013a. Towards domain adaptation for parsing web data. In *Proceedings of the International Conference Recent Advances in Natural*

[32] To address the variety of genres across Old and Modern Italian, we need annotated data for both non-poetic Old Italian literature and Modern Italian poetry, currently unavailable.

Language Processing RANLP 2013, pages 357–364, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Mohammad Khan, Markus Dickinson, and Sandra Kuebler. 2013b. Does size matter? text and grammar revision for parsing social media data. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 1–10, Atlanta, Georgia. Association for Computational Linguistics.

Francesco Mambrini and Marco Carlo Passarotti. 2012. Will a parser overtake Achilles? First experiments on parsing the ancient Greek dependency treebank. In *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories (TLT11). 30 November–1 December 2012, Lisbon, Portugal*, pages 133–144. Edições Colibri.

Paola Manni. 2013. *La lingua di Dante*. il Mulino, Bologna, Italy.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. pages 101–108.

Giampaolo Salvi and Lorenzo Renzi, editors. 2010. *Grammatica dell'italiano antico*. il Mulino, Bologna, Italy.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Mirko Tavoni. 2011. *DanteSearch: il corpus delle opere volgari e latine di Dante lemmatizzate con marcatura grammaticale e sintattica*, volume 2 (2004–2005), pages 583–608. Il Torcoliere – Officine Grafico-Editoriali di Ateneo, Napoli, Italy.

Sara Tonelli, Rodolfo Delmonte, and Antonella Bristot. 2008. Enriching the venice Italian treebank with dependency and grammatical relations. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Unsupervised Authorship Attribution for Medieval Latin using Transformer-Based Embeddings

Loic De Langhe, Orphée De Clercq, Veronique Hoste
LT3, Language and Translation Technology Team, Ghent University, Belgium
Groot-Brittanniëlaan 45, 9000 Ghent, Belgium
firstname.lastname@ugent.be

Abstract

We explore the potential of employing transformer-based embeddings in an unsupervised authorship attribution task for medieval Latin. The development of Large Language Models (LLMs) and recent advances in transfer learning alleviate many of the traditional issues associated with authorship attribution in lower-resourced (ancient) languages. Despite this, these methods remain heavily understudied within this domain. Concretely, we generate strong contextual embeddings using a variety of mono -and multilingual transformer models and use these as input for two unsupervised clustering methods: a standard agglomerative clustering algorithm and a self-organizing map. We show that these transformer-based embeddings can be used to generate high-quality and interpretable clusterings, resulting in an attractive alternative to the traditional feature-based methods.

Keywords: Authorship Attribution, Medieval Latin, Unsupervised Learning

1. Introduction

Throughout modern history, scholars have always been greatly interested in the authenticity and authorship of important historical documents. In the fifteenth century, Renaissance scholar Lorenzo Valla exposed the purported 4th century imperial decree *Donatii Constantini* as an 8th century forgery by comparing the language in the document with actual 4th century Latin sources. A little over 500 years later, Mosteller and Wallace (1963) showed through statistical analysis that James Madison, rather than Alexander Hamilton, was the author of 12 disputed documents in the (in)famous *Federalist papers*. In short, the methods may have changed, but the question has remained the same.

In a computational setting, authorship analysis is often analogous to stylometry i.e. the use of quantifiable and statistical methods to unmask an author's stylistic DNA or signature (Holmes, 1998). At the forefront of this field lies the idea that individual authors have a marked and highly specific writing style that can be used to separate them from others (Stamatatos, 2009). Modern stylometric studies typically focus on the attribution of essays, emails and forum posts to distinct online users or groups of users (Kestemont et al., 2018). Naturally, the field ties in to modern-day applications such as plagiarism detection, identity deception on social media platforms and multi-modal authentication on mobile devices (Neal et al., 2017). While there is an emphasis on applying stylometric methods in modern settings, the stylistic analysis of work from the antiquity and medieval periods also remains a highly studied topic. The emergence and distribution of large electronic document collections containing heaps of anonymous or (seemingly) miss-attributed texts has lead to many researchers continuing directly in Lorenzo Valla's footsteps, more than 500 years after his passing.

Research on antique and medieval texts is hampered by a general lack of spelling and language standardization as well as transcription errors (Kestemont, 2012). This naturally poses an additional layer of difficulty, as it is hard to determine whether or not the spelling of a word is due to the original author's stylistic signature, or was introduced by those transcribing the work. Nonetheless, computational stylometric analysis of antique and medieval texts has led to the identification of previously anonymous authors, or the rectification of the authorship of misattributed work (Stover et al., 2016; Kabala, 2020). It is to be noted that, unlike in most NLP domains, the use of neural approaches remains limited, mostly due to the lack of large amounts of training data, which these deep neural architectures typically require to function optimally (Corbara et al., 2023). Nonetheless, recent advancements in the field of Natural Language Processing (NLP) have given rise to large-scale transformer architectures which circumvent the need for large task-specific corpora through transfer learning. Despite their ability to capture accurate representations of longer documents and encode implicit textual structures, transfer learning methods remain understudied in the context of medieval stylometry.

In this paper, we explore the potential of using a variety of transformer-based models for unsupervised authorship attribution in Medieval Latin. Concretely, we generate powerful vectorial representations of Medieval Latin texts and use these as a basis for two unsupervised clustering methods:

a standard agglomerative clustering algorithm and a self-organizing map (SOM). The former serves as our primary method for intrinsic and extrinsic evaluation of the generated clusters, while the latter aims to create highly interpretable visualisations of the data. We show that, without relying on a series of highly specialized manually crafted features, we can accurately cluster a large number of 13th-14th century Latin texts by author, illustrating the potential of using transfer-learning methods in future stylometric studies.

2. Related Work

Work on computational methods for authorship attribution goes back to the very beginning of the field of Computational Linguistics (CL) as a whole (Holmes, 1998). Earlier work often focused on well-known contested English texts, with the disputed *Federalist Papers* being a notable example that has been studied multiple times throughout the years (Mosteller and Wallace, 1963; Tweedie et al., 1996). More recently however, there has been a growing interest in performing computational stylistic analysis on a wider range of languages such as Dutch (Kestemont, 2012; Morante et al., 2022), Ancient Greek (Gorman and Gorman, 2016), Spanish (López-Escobedo et al., 2013) and many others (Savoy, 2020).

For Latin specifically, there have been, among others, stylometric studies regarding the works of Hildegard of Bingen (Kestemont et al., 2015), Dante Alighieri (Corbara et al., 2019) and the attribution of a newly discovered manuscript to the writer Apuleius (Stover et al., 2016). Additionally, specific authorship attribution tools such as *Medievalla* have been developed and made available to the wider research community (Corbara et al., 2022). Note that most of these studies largely follow the same approach: the combination of rigorously handcrafted stylistic features combined with traditional machine learning algorithms (Muldoon et al., 2021). While there have been recent studies that combine well-known stylistic markers such syllabic patterns with deep neural networks (Corbara et al., 2023), more modern neural methods such as transformer-based architectures remain a largely unexplored approach.

All of the methods earlier described made use of the standard supervised learning paradigm in which the ground truth (or gold-standard labeling) is known and used to evaluate the performance of a given algorithm. Nonetheless, unsupervised approaches are often being applied to (historical) NLP tasks to automatically find underlying patterns without the need for human intervention (Kehler and Stolcke, 1999; Bharadiya, 2023). For authorship attribution specifically clustering algorithms are often applied to uncover implicit similarity between the works of known writers and anonymous documents or to determine outliers (i.e. possibly misattributed works) in their bibliography (Martín-del Campo-Rodríguez et al., 2022). Research on unsupervised methods for stylometry often makes use of the popular agglomerative clustering algorithm (Layton et al., 2013; Panicheva et al., 2019), but other methods such as c-means (Demir, 2013) and self-organizing maps (Ranatunga et al., 2011; Neme et al., 2015) have also been applied. Note also that most studies involving unsupervised learning forgo the use of hand-crafted feature sets and instead focus on more easily extractable textual information such as character n-grams (Kapočiūtė-Dzikienė et al., 2015), punctuation (Tanguy et al., 2012) or rudimental similarity functions between texts (Qian et al., 2015).

3. Experiments

3.1. Data

Our data consists of the Medlatin1 and Medlatin2 corpora, which are composed of 13-14th century Latin epistles (MedLatin1) and literary analyses (Medlatin2) by a variety of authors (Corbara et al., 2022). As was done in Corbara et al. (2023), we merge the two corpora resulting in one dataset encompassing 324 medieval Latin texts. We then remove a total of 31 epistels for which no specific author is known, resulting in a final collection of 293 documents.

3.2. Experimental Setup

3.2.1. Agglomerative Clustering

First, we apply an agglomerative clustering algorithm which uses an average linkage criterion i.e. two clusters are merged based on the average of distances between all pairs of both objects. For two clusters A and B the distance between them is defined as:

$$d_{AB} = \frac{1}{kl} \sum_{i=1}^{k} \sum_{j=1}^{l} d(X_i, Y_j)$$

Where X_i and Y_j are objects within clusters A and B respectively and $d(.)$ is the distance (cosine) function. The results of this algorithm will serve as our prime (numerical) evaluation of cluster quality. Note that unsupervised methods are typically evaluated both intrinsically (unsupervised, cluster quality and how well the clusters are separated) and extrinsically (supervised, based on the gold-standard labels). For our analysis we will take both evaluation strategies into account.

3.2.2. Self-Organizing Map

In addition to the standard clustering algorithm, we train a self-organizing map (SOM) neural network, which will allow a more interpretable analysis of the obtained clusters. The self-organizing map (Oja and Kaski, 1999) is a 2-dimensional representation of a series of data points which respects the topological structure of the dataset. We follow the standard SOM algorithm as it was presented in Oja and Kaski (1999). First, a document x is sampled randomly from the collection and based on the randomly initialized weights w of the neurons in the lattice the best matching unit (BMU) is determined:

$$i(x) = argmin_j \, \|w - w_j\|$$

The weights in the lattice are then updated through a Hebbian learning rule where η is the learning rate and $h(j, i(x))$ is the (Gaussian) neighborhood function which allows incremental updates to neurons surrounding the BMU:

$$w_j \leftarrow w_j + \eta h(j, i(x)))(x - w_j)$$

3.2.3. Textual representation

For both methods we present each individual document in the dataset as a transformer-generated representation of said document. Each text is passed through a transformer encoder to create a high-dimensional vector representation (embedding). Following earlier studies on the effectiveness of using transformer-based embeddings (Devlin et al., 2018), we generate document embeddings based on several encoder layers, rather then only using the last layer as an instance's representation. We concatenate the transformers' last four encoder layers (each a vector of length 768) to a 3072-dimensional feature representation for each document. We compare four distinct models in order to broadly gauge their capabilities w.r.t medieval Latin. First, a monolingual Latin RoBERTa model[1] which was trained on the Latin part of the cc-100 corpus (Conneau et al., 2019). Second, a multilingual encoder model which was trained on a total of 104 languages (including Latin) (Devlin et al., 2018). Third, a multilingual model using the DeBERTaV3 architecture (He et al., 2021), which has been shown to outperform most monolingual models in a large variety of languages. The final model tested in our experiments is a longformer model. Most BERT-based encoders suffer from processing longer texts as the token limit of an input is restricted to 512. Longformer-inspired models however use a linearly scaling attention mechanism which poses significantly less strain on computational resources and allows processing of sequences of up to 4096 tokens (Beltagy et al., 2020). Given the fact that many texts of the Medlatin1 and Medlatin2 corpora are quite lengthy, long-document transformers may be more suited. The multilingual longformer model used in the experiments was trained on 103 languages (including Latin) of the cc-100 corpus [2].

3.3. Hardware and Software Implementation

All experiments were trained and evaluated on a single Tesla V100-SXM2-16GB GPU. For the implementation of the agglomerative clustering algorithm we relied on the use of Python's Scikit-Learn module (Pedregosa et al., 2011). The training and visualisation of the SOM algorithm was performed through the MiniSom package [3]. Specific training parameters can be found in Appendix A.

4. Results

4.1. Agglomerative Clustering

Most unsupervised clustering algorithms are evaluated through intrinsic methods, which evaluate the quality of a clustering by how well the clusters are separated. For this paper, we evaluate the generated clusterings through two intrinsic measures, which both measure how similar an object is to its own cluster compared to other clusters: the silhouette coefficient (SC), which ranges from -1 to +1 with higher values indicating better clusterings and the Calinski-Harabasz Index (CHI), the value of which is unrestricted and for which higher values indicate higher quality clusters. In addition to these intrinsic measures we also include the Rand Index (RI) as an evaluation metric, which computes the degree of similarity between two data partitions (the predictions and the ground truth). This metric ranges from 0 to 1, with higher values indicating larger similarity between the generated clusters and the gold standard. Table 4.1 contains the results for each of the clusters generated through the embeddings of the different encoder models.

Model	SC	CHI	RI
Longformer	0.3975	**197.00**	**0.7382**
mBERT	**0.4796**	34.77	0.6672
RoBERTa Latin	0.2526	25.01	0.6708
mDeBERTaV3	0.3036	26.41	0.6155

Table 1: Silhouette Coefficient (SC), Calinski-Harabasz Index (CHI) and Rand Index (RI) scores for the generated clusterings.

[1] https://huggingface.co/pstroe/roberta-base-latin-cased

[2] https://huggingface.co/markussagen/xlm-roberta-longformer-base-4096

[3] https://github.com/JustGlowing/minisom

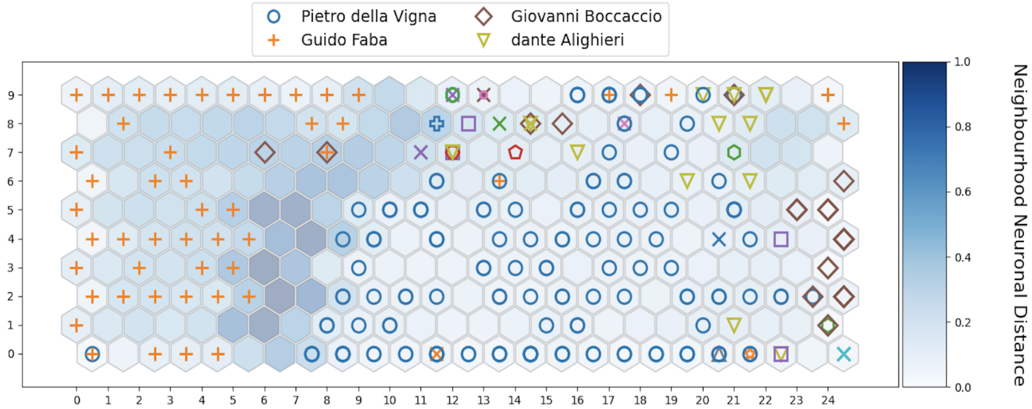


Figure 1: Visualisation of the trained self-organizing map using the Longformer embeddings as document representations.

Overall, we find that clusters generated through the embeddings of the longformer-xlmr model performed best on average, both by means of intrinsic and extrinsic evaluation. We hypothesize here that the signficantly larger context length of 4096 tokens (as opposed of 512 for the other models) ultimately plays a significant role in capturing an author's stylistic signature. We also note that the obtained RI scores for each of the models can be interpreted as moderate-to-high overlap between the generated clusters and the ground truth, indicating that unsupervised clustering through transfer learning may be a viable method for large scale analysis in the future. Interestingly, while the monolingual Latin model shows comparatively good results for the extrinsic evaluation, the intrinsic evaluation is significantly worse than the other models. This can indicate that the generated clusters, while distinguishable to a degree, are highly similar to one another. In the context of this task, this means that the authors' stylistic signatures are captured comparatively less by the monolingual Latin model.

4.2. Self-Organizing Map

We obtain a detailed topological map of the data by initializing the SOM with a 10-by-25 lattice and training the algorithm using the learning rules described in Section 3.2.2. The resulting topological map using the best model embeddings (longformer) can be seen in Figure 1. For readability's sake the legend in Figure 1 only includes the 4 most represented authors of the dataset which are (in order): Pietro Della Vigna (n = 146), Guido Faba (n = 78), Giovanni Boccaccio (n = 27) and Dante Alighieri (n = 14). A detailed legend of all 22 authors as well as the topological representations generated with the other three encoder models can be found in Appendix B.

We do not rely on quantitative metrics for the evaluation of the generated lattice, but rather on visual analysis. We observe that the SOM presents a qualitative clustering of the various authors, with the four most prominent authors clearly occupying four distinct spaces on the map. Note that the works of Guido Faba are seen as highly distinct from the other works in the dataset. Interestingly, one particular letter by Pietro Della Vigna is significantly closer to the letters of Guido Faba than to della Vigna's other works. In the end, only close reading and study can ultimately provide clarity regarding the authorship of unattributed or dubious manuscripts. Nonetheless, the identification of outliers, such as the one mentioned, through unsupervised computational analysis can serve as an early diagnostic step in this process as well as narrowing the scope of this complex task.

Finally, we also observe that for some authors with only one work in the dataset, the neuronal distance to neighboring positions is remarkably high. This indicates that the SOM neural network can segment individual authors' stylistic signatures even if there is only a limited amount of their work available. In this way, the SOM algorithm can be an effective way to detect outliers within larger document collections. This is a notable advantage of applying a SOM compared to more traditional clustering methods, which often continuously merge clusters until an arbitrary threshold is reached and thus concentrate less on the uniqueness of individual data points.

5. Conclusion

We show for the first time that transformer-generated contextual embeddings can be used to render qualitative unsupervised clusterings of author attributions in medieval Latin. We examined the embeddings of four distinct transformer models and found, through both intrinsic and extrinsic evaluation, that long-document transformer models lead to the best available clusterings. While close-reading and traditional feature-based methods are still needed to conclusively determine the authenticity or attribution of (ancient) manuscripts, we believe that transfer learning methods can be used as an early diagnostic tool for both outlier detection and narrowing the search space within large medieval document collections.

6. Acknowledgements

This work was supported by the Research Foundation–Flanders under project grant number FWO.OPR.2020.0014.01.

7. Bibliographical References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Jasmin Bharadiya. 2023. A comprehensive survey of deep learning techniques natural language processing. *European Journal of Technology*, 7(1):58–66.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Silvia Corbara, Alejandro Moreo, and Fabrizio Sebastiani. 2023. Syllabic quantity patterns as rhythmic features for latin authorship attribution. *Journal of the Association for Information Science and Technology*, 74(1):128–141.

Silvia Corbara, Alejandro Moreo, Fabrizio Sebastiani, and Mirko Tavoni. 2019. The epistle to cangrande through the lens of computational authorship verification. In *New Trends in Image Analysis and Processing–ICIAP 2019: ICIAP International Workshops, BioFor, PatReCH, e-BADLE, DeepRetail, and Industrial Session, Trento, Italy, September 9–10, 2019, Revised Selected Papers 20*, pages 148–158. Springer.

Silvia Corbara, Alejandro Moreo, Fabrizio Sebastiani, and Mirko Tavoni. 2022. Medlatinepi and medlatinlit: Two datasets for the computational authorship analysis of medieval latin texts. *Journal on Computing and Cultural Heritage (JOCCH)*, 15(3):1–15.

Nesibe Merve Demir. 2013. Artificial neural network techniques in authorship attribution. *Southeast Europe Journal of Soft Computing*, 2(2).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Vanessa B Gorman and Robert J Gorman. 2016. Approaching questions of text reuse in ancient greek using computational syntactic stylometry. *Open Linguistics*, 2(1).

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

David I Holmes. 1998. The evolution of stylometry in humanities scholarship. *Literary and linguistic computing*, 13(3):111–117.

Jakub Kabala. 2020. Computational authorship attribution in medieval latin corpora: the case of the monk of lido (ca. 1101–08) and gallus anonymous (ca. 1113–17). *Language Resources and Evaluation*, 54(1):25–56.

Jurgita Kapočiūtė-Dzikienė, Andrius Utka, and Ligita Šarkutė. 2015. Authorship attribution of internet comments with thousand candidate authors. In *Information and Software Technologies: 21st International Conference, ICIST 2015, Druskininkai, Lithuania, October 15-16, 2015, Proceedings 21*, pages 433–448. Springer.

Andrew Kehler and Andreas Stolcke. 1999. Unsupervised learning in natural language processing. In *Association for Computational Linguistics. Proceedings of the workshop. In Preface A. Kehler and A. Stolcke, editors*.

Mike Kestemont. 2012. Stylometry for medieval authorship studies: an application to rhyme words. *Digital Philology: A Journal of Medieval Cultures*, 1(1):42–72.

Mike Kestemont, Sara Moens, and Jeroen Deploige. 2015. Collaborative authorship in the twelfth century: A stylometric study of hildegard of bingen and guibert of gembloux. *Digital Scholarship in the Humanities*, 30(2):199–224.

Mike Kestemont, Michael Tschuggnall, Efstathios Stamatatos, Walter Daelemans, Günther Specht, Benno Stein, and Martin Potthast. 2018. Overview of the author identification task at pan-2018: cross-domain authorship attribution and style change detection. In *Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018/Cappellato, Linda [edit.]; et al.*, pages 1–25.

Robert Layton, Paul Watters, and Richard Dazeley. 2013. Automated unsupervised authorship analysis using evidence accumulation clustering. *Natural Language Engineering*, 19(1):95–120.

Fernanda López-Escobedo, Carlos-Francisco Méndez-Cruz, Gerardo Sierra, and Julián Solórzano-Soto. 2013. Analysis of stylometric variables in long and short texts. *Procedia-Social and Behavioral Sciences*, 95:604–611.

Carolina Martín-del Campo-Rodríguez, Grigori Sidorov, and Ildar Batyrshin. 2022. Unsupervised authorship attribution using feature selection and weighted cosine similarity. *Journal of Intelligent & Fuzzy Systems*, 42(5):4357–4367.

Roser Morante, Eleanor LT Smith, Lianne Wilhelmus, Alie Lassche, and Erika Kuijpers. 2022. Identifying copied fragments in a 18th century dutch chronicle. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5865–5878.

Frederick Mosteller and David L Wallace. 1963. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers. *Journal of the American Statistical Association*, 58(302):275–309.

Connagh Muldoon, Ahsan Ikram, and Qublai Ali Khan Mirza. 2021. Modern stylometry: A review & experimentation with machine learning. In *2021 8th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 293–298. IEEE.

Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. 2017. Surveying stylometry techniques and applications. *ACM Computing Surveys (CSuR)*, 50(6):1–36.

Antonio Neme, JRG Pulido, Abril Muñoz, Sergio Hernández, and Teresa Dey. 2015. Stylistics analysis and authorship attribution algorithms based on self-organizing maps. *Neurocomputing*, 147:147–159.

Erkki Oja and Samuel Kaski. 1999. *Kohonen maps*. Elsevier.

Polina Panicheva, Olga Litvinova, and Tatiana Litvinova. 2019. Author clustering with and without topical features. In *International Conference on Speech and Computer*, pages 348–358. Springer.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Tie-Yun Qian, Bing Liu, Qing Li, and Jianfeng Si. 2015. Review authorship attribution in a similarity space. *Journal of Computer Science and Technology*, 30(1):200–213.

RVSPK Ranatunga, AS Atukorale, and KP Hewagamage. 2011. Intrinsic plagiarism detection with kohonen self organizing maps. In *U The International Conference on Advances in ICT for Emerging Regions-ICTer2011*, volume 125.

Jacques Savoy. 2020. Machine learning methods for stylometry. *Cham: Springer*.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556.

Justin Anthony Stover, Yaron Winter, Moshe Koppel, and Mike Kestemont. 2016. Computational authorship verification method attributes a new work to a major 2nd century a frican author. *Journal of the Association for Information Science and Technology*, 67(1):239–242.

Ludovic Tanguy, Franck Sajous, Basilio Calderone, and Nabil Hathout. 2012. Authorship attribution: Using rich linguistic features when training data is scarce. In *PAN Lab at CLEF*.

Fiona J Tweedie, Sameer Singh, and David I Holmes. 1996. Neural network applications in stylometry: The federalist papers. *Computers and the Humanities*, 30:1–10.

A. Appendix A

Parameter	Value
Lattice Dimension	10x25
Learning Rate	0.7
Neighborhood Function	Gaussian
Distance Metric	Cosine Distance
Topology Configuration	Hexagonal
Neighborhood Radius	6
Training Iterations	1000

Table 2: Training configuration for the SOM algorithms. All SOM representations were trained using identical parameters.

B. Appendix B

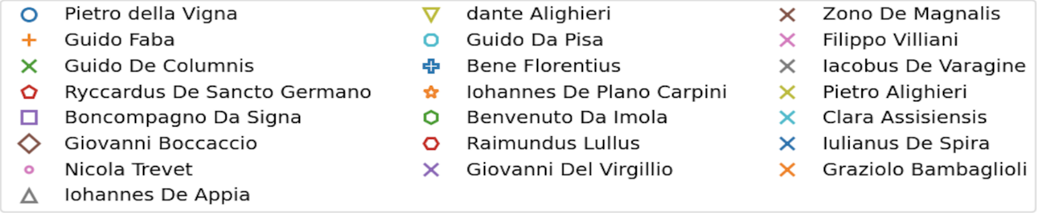


Figure 2: Complete legend of all 22 authors for the SOM visualisations.

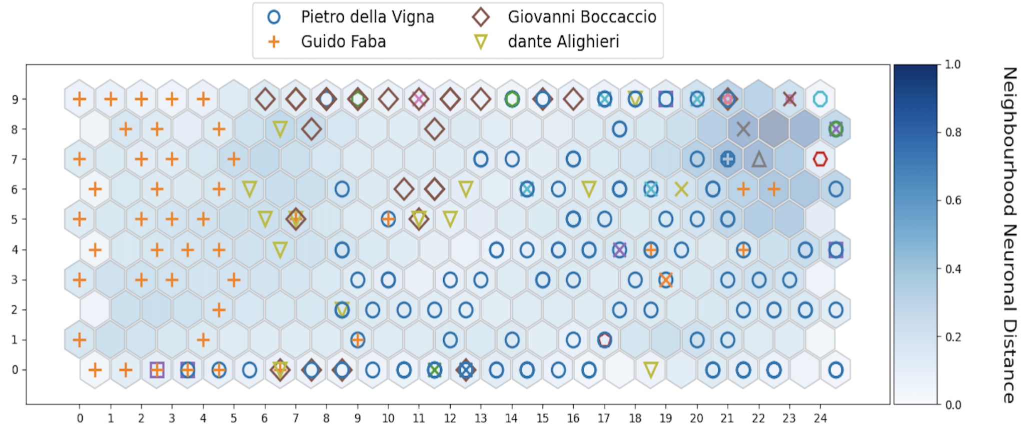


Figure 3: Visualisation of the trained self-organizing map using the mDeBERTaV3 embeddings as document representations.

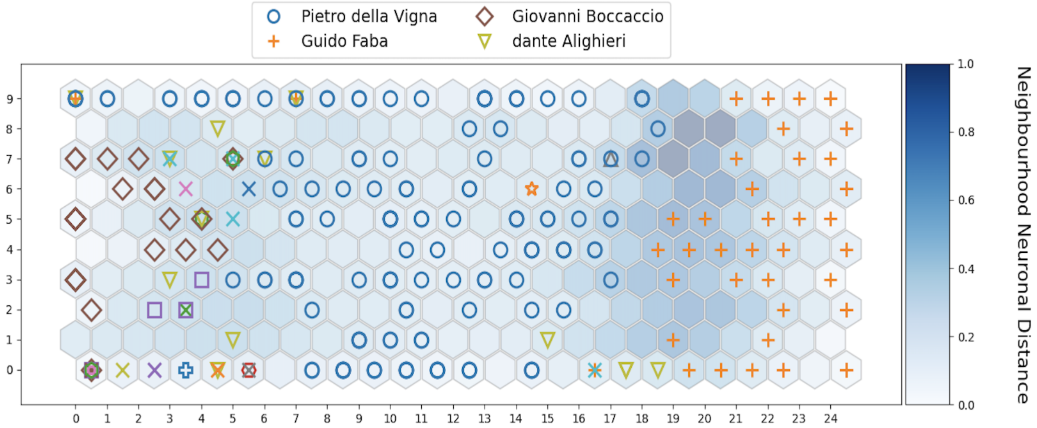


Figure 4: Visualisation of the trained self-organizing map using the Latin RoBERTa embeddings as document representations.

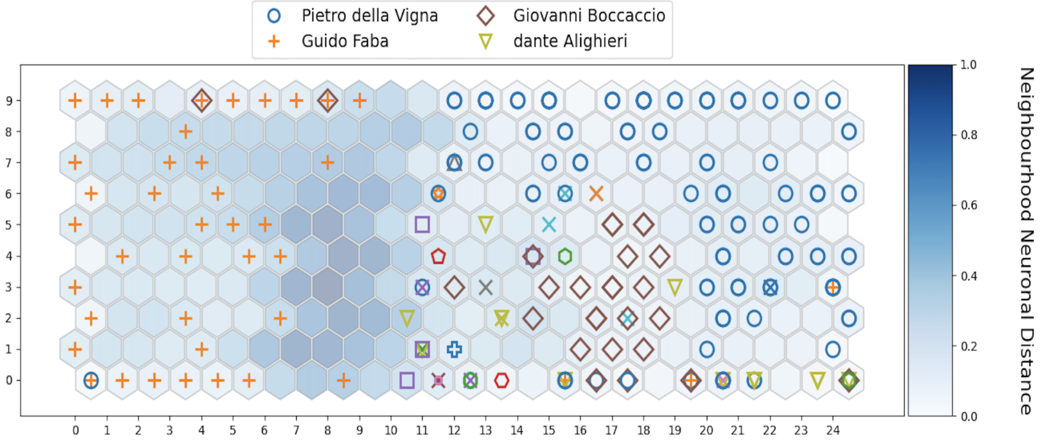


Figure 5: Visualisation of the trained self-organizing map using the mBERT embeddings as document representations.

“To Have the ‘Million’ Readers Yet”: Building a Digitally Enhanced Edition of the Bilingual Irish-English Newspaper *An Gaodhal* (1881 – 1898)

Oksana Dereza[1,2], **Deirdre Ní Chonghaile**[3], **Nicholas Wolf**[3,4]
[1] University of Galway Library, Ireland
[2] Insight SFI Centre for Data Analytics, Data Science Institute, University of Galway, Ireland
[3] Glucksman Ireland House, New York University, USA
[4] Division of Libraries, New York University, USA
oksana.dereza@universityofgalway.ie, nichonghailed@ollscoilnagaillimhe.ie, nicholas.wolf@nyu.edu

Abstract

This paper introduces two new OCR models for the Irish language, a BART-based OCR post-correction model, and the core dataset on which they were trained: a monthly bilingual Irish-English newspaper named *An Gaodhal* that was produced from 1881 to 1898 by an Irishman living in Brooklyn, New York.

Keywords: Optical Character Recognition (OCR), OCR post-correction, Named Entity Recognition (NER), Irish, Gaeilge, bilingual data, code-mixing, digital edition, corpus creation, corpus annotation

1. Introduction

This paper introduces the *An Gaodhal* project, which aims to serve the historically under-resourced and endangered language of Irish[1] (known as Gaeilge) by providing new digital tools and resources.

The initial goal of the project was the extraction of full text of *An Gaodhal*, a monthly bilingual Irish-English newspaper produced from 1881 to 1898 by an Irishman living in Brooklyn, New York, to the highest possible degree of accuracy via Optical Character Recognition (OCR), with a view to making its printed content searchable. The methodology applied toward achieving this goal yielded additional digital outputs including:

- a new OCR model for the Irish language as printed in Cló Gaelach type;[2]
- a new OCR model for bilingual Irish-English content printed in Cló Gaelach and Roman types respectively;
- a BART-based OCR post-correction model for historical bilingual Irish-English data;
- a historical Irish training set for Named Entity Recognition (NER);

All but the first of these four additional outputs appear to be the first of their kind. Each of the project outputs is set for public release to enable open-access research.

This paper also identifies the challenges historical Irish data poses to Natural Language Processing (NLP) in general and OCR in particular, and reports on project results and outputs to date. Finally, it contextualises the project within the wider field of NLP and considers its potential impact on under-resourced languages worldwide.

2. Related Work

2.1. OCR

In December 2022, the Irish government launched a roadmap document titled *Digital Plan for the Irish Language: Speech and Language Technologies 2023-2027*, which “provides an overview of the research required to make Irish-language linguistic resources available in the coming years” (Government of Ireland, 2022). This digital plan acknowledges the need for a diverse ecosystem of Irish-language corpora and identifies a significant number already extant for Irish. To date, only one of these corpora — *Corpas Stairiúil na Gaeilge 1600 – 1926* (Acadamh Ríoga na hÉireann, 2017) — has been produced using OCR. To the best of our knowledge, the relevant OCR work was outsourced to a third-party, and any models deployed in that work were not available publicly.

At the outset of the present project in January 2023, there were no publicly available OCR models attuned to Cló Gaelach and pre-standardised spelling of the Irish language, in either monolingual or multilingual contexts.[3] The only related project in existence was a Cló Gaelach training dataset for Tesseract OCR software,[4] published by Scannell et al. (2020). In November 2023, an

[1] Moseley (2010) lists Irish as ‘definitely endangered’.

[2] Cló Gaelach – a typeface widely used for Irish until the 1960s when it was replaced by Roman type.

[3] In a bilingual / multilingual context, Irish appears most frequently alongside English, reflecting their co-existence in Ireland for centuries.

[4] https://github.com/tesseract-ocr/tesseract

Irish-only model for texts in either Cló Gaelach or Roman typefaces was made public on the Transkribus OCR platform (Farrell, 2023). The methodology that produced this model differs considerably from the approach discussed herein in respect of the treatment of different typefaces, the treatment of individual printed glyphs, and the broader span of centuries represented in the corpus upon which the model was trained.

OCR models vary enormously, ranging from bespoke monolingual models, some of them reading individual handwritten scripts, to large-scale models incorporating multiple languages. Transkribus Team (2021) created the multilingual multi-typeface print model *Transkribus Print M1*, "including antiqua and blackletter prints, typewriter, computer print outs and decorative fonts" and supporting Dutch, English, Finnish, French, German, Italian, Latin, Swedish, Portuguese, Spanish, Polish, Flemish, Czech, Slovak, Slovenian, and Castilian. Currently, Transkribus features 147 publicly available models for print and handwritten text recognition,[5] including Devanagari, Hebrew, Ethiopian script, 14^{th} and 15^{th} century Spanish Gothic script, 14^{th} century cursive Dutch charters, 16^{th} century Balinese palm-leaf manuscripts, Serbian and Russian Church Slavonic, Ottoman Turkish written in Arabic script, 19^{th} century Danish handwriting, and multiple varieties of Fraktur[6] to name but a few.

Some OCR models focus on individual ancient and historical languages (Furrer and Volk, 2011; Bukhari et al., 2017; Springmann et al., 2018; Reul, 2020; Reul et al., 2021; Martínek et al., 2020; Dölek and Kurt, 2022; Ma et al., 2024). Others address multilinguality in a historical context: a team at Cornell University developed a trilingual handwritten text recognition (HTR) model for Ancient Greek, Latin, and German (Rusten, 2020);[7] and Capurro et al. (2023) are testing the viability of different approaches to building multilingual OCR models for HTR. A significant share of research on pre-modern OCR draws on historical newspaper corpora (Drobac et al., 2017; Koistinen et al., 2020; Drobac, 2020; Kettunen et al., 2020), which are readily accessible thanks to trends in early institutional digitisation.

Predictably, OCR datasets that predate the emergence of more advanced technologies register higher error rates. Moreover, source images are not always retained. The resulting impossibility or cost of re-extracting text prompts researchers to explore OCR post-correction as a discrete task (Reynaert, 2008; Vobl et al., 2014; Reynaert, 2016; Afli et al., 2016; Schulz and Kuhn, 2017; Richter et al., 2018; Dong and Smith, 2018; Dannélls and Persson, 2020; Duong et al., 2021; Soper et al., 2021; Rijhwani et al., 2021; Besnier and Mattingly, 2021; Lyu et al., 2021; Suissa et al., 2022). OCR post-correction is also applied to critically endangered languages where a scarcity of data would otherwise impede the building of a targeted OCR model. In such cases, scholars train a correction model to transform outputs of an OCR model unfamiliar with the target language (Rijhwani et al., 2020), a method that ultimately aims to obtain the best OCR results for the target language.

The capacity of OCR to inspire "new kinds of research on previously inaccessible sources" in humanities and social sciences is driving unprecedented growth in this domain and scholars continue to explore ways of improving OCR outputs (Smith and Cordell, 2018).

2.2. NER

Like OCR, NER work around the globe reflects a wide variety of approaches, some of which are discussed in an extensive survey on NER in historical documents published last year (Ehrmann et al., 2023). To date, two shared tasks on "identifying historical people, places and other entities (HIPE)" have been organised (Ehrmann et al., 2020, 2022). Some NER work focuses on individual historical languages, including 19^{th} century French (Tual et al., 2023), 8^{th} century Armenian (Tambuscio and Andrews, 2021), and Ancient Greek (Yousef et al., 2023); some aims at developing multilingual NER models (Neudecker, 2016; Boros et al., 2020; Dekhili and Sadat, 2020; Provatorova et al., 2020; Schweter et al., 2022). Like the *An Gaodhal* project, some teams combine OCR, OCR post-correction, and NER in historical texts (Todorov and Colavizza, 2020). As with OCR, historical newspaper data is well-represented in NER research (Hubková, 2019; Schweter and Baiter, 2019; Hubková et al., 2020),

NER for the Irish language, whether modern or historical, represents uncharted territory. According to the Government of Ireland (2022): "To date, there is no named-entity recognition system available for Irish. There are some basic resources (lists of named entities) available through the part-of-speech tagger technology, and place names at `logainm.ie` but much work is required to extend this research into a comprehensive NER tool."

3. Historical Context

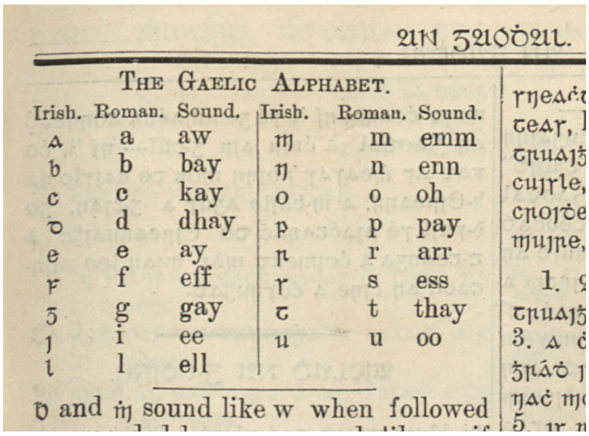
Historically, the Irish language has been printed in two different orthographies: Irish or Gaelic type,

[5] https://readcoop.eu/transkribus/public-models/

[6] *Fraktur* denotes the German blackletter, or 'Gothic', fonts that derive from medieval handwriting.

[7] The authors could not locate this model on Transkribus, and assume it has not been made public.

known as Cló Gaelach, which originated in the scribal tradition (see Figure 1); and Roman type (McGuinne, 1992). Its corresponding Unicode characters draw on Roman (Latin) script. Cló Gaelach uses two kinds of diacritics: acute accents on vowels (ÁáÉéÍíÓóÚú); and dotted consonants (ḂḃĊċḊḋḞḟĠġṀṁṖṗṠṡṪṫ), the dots indicating a grammatical feature called lenition. Where Irish appears in Roman type, dotted consonants are replaced by Bh, Ch, dh, fh, etc.



AN GAOḊAL.

THE GAELIC ALPHABET.

Irish.	Roman.	Sound.	Irish.	Roman.	Sound.
a	a	aw	m	m	emm
b	b	bay	n	n	enn
c	c	kay	o	o	oh
d	d	dhay	p	p	pay
e	e	ay	r	r	arr
f	f	eff	s	s	ess
g	g	gay	t	t	thay
i	i	ee	u	u	oo
l	l	ell			

ḃ and ṁ sound like w when followed

sneacṫ
tear,
truaiġ
cuisle,
croiḋe
muire,
1.
truaiġ
3. a ċ
ġráḋ
nac m
5.

Figure 1: The Irish alphabet.

Although the quantity of printed material in Irish in the centuries prior to the appearance of *An Gaodhal* in October 1881 was small in comparison to many languages, a recent cataloguing of titles published in Irish between the 16^{th} and 19^{th} centuries (Sharpe and Hoyne, 2020) lists over a thousand entries, several of them with multiple editions. Prior to the 1880s, the most common genres for printing in Irish were religious texts (both Catholic and Protestant), academic texts, and so-called Gaelic columns in otherwise English-only newspapers in which a relatively small amount of content (usually letters, songs, or poetry) was printed in Irish. *An Gaodhal* thus appeared at a time when printing in Irish was taking place, but not on a mass scale, so the newspaper's production represented an energetic undertaking in the face of headwinds.

An Gaodhal was established and edited by Micheál Ó Lócháin (also known as Michael J. Logan).[8] It is regarded as the world's first serial dedicated to providing content to an Irish-language readership. The first four issues of the newspaper were printed commercially and at a loss. To save the enterprise, Logan took on the task of typesetting and printing the newspaper himself, most likely in his own home in Brooklyn. Over the next 17 years, Logan continued to issue the paper, supported by a transnational network of contributors. His commitment combined with the appetite among readers to achieve 1,200 subscriptions within the first year, growing to 3,000 at its peak, five times the number achieved by the contemporaneous Dublin-based *Irisleabhar na Gaedhilge*, also known as *The Gaelic Journal* (Uí Fhlannagáin, 1990).

As one might expect from an ethnic newspaper emerging in a diasporic setting, contributors to *An Gaodhal* and its readers welcomed the arrival of a new forum in which to identify their community of 'Éire Mhór' (Greater Ireland) and celebrate it (Knight, 2021). Nationalist politics at home in Ireland amplified that sense of pride, which extended to the use of Cló Gaelach throughout the newspaper to distinguish Irish expression from the English nation, its language and Roman type, and British imperialism. Indeed, the Irish type used in the newspaper, modelled on Watts type, was newly cast in the United States to avoid purchasing a set cast in a London foundry.

There is a palpable sense of energy and excitement in the newspaper as many of its contributors and readers were then gaining literacy in Irish for the first time. The standard of written Irish varied accordingly, as did the spelling, which had yet to be standardised. Add to this the use of three differing dialects and the emerging corpus of texts — however small at 1.86 million tokens — yields a welcome diversity in the prospective training data. To date, the adaptability of the OCR models developed by the project team supports this inference.

The challenges *An Gaodhal* faced were varied. The economics of audience size over printing costs, particularly for a newspaper printed for a transatlantic audience, drove its founder and editor to forgo any income from his work on the paper. The debate over the choice of type, whether Roman or Cló Gaelach, had long been a heated one; for those who insisted that Cló Gaelach was the only proper type for expressing Irish, there was the immediate challenge of procuring such a unique typeface — a matter of availability, not cost, as it could be purchased for the same price as Roman type. Even where Roman type was selected, any printer choosing to produce Irish texts in the nineteenth century or earlier faced difficulties in finding a sufficiently large, paying audience and, in a diasporic context, sufficiently fluent typesetters or compositors. The absence of mass literacy in Irish prior to the twentieth century combines with these challenges for printing to make the appearance of *An Gaodhal* and of similar undertakings in its aftermath[9] especially notable: they represent

[8] See https://www.ainm.ie/Bio.aspx?ID=347 (Irish) and https://www.dib.ie/biography/logan-michael-j-o-lochain-micheal-a4873 (English) for Michael J. Logan's biographies.

[9] See, for example, Knight (2021) on the Irish-language column in *New York Irish-American*, 1857 –

the first steps in creating a media landscape in the Irish language, an impact foretold in the ambition expressed by Logan in *An Gaodhal* “to have the ‘million’ readers yet.”[10]

4. Data

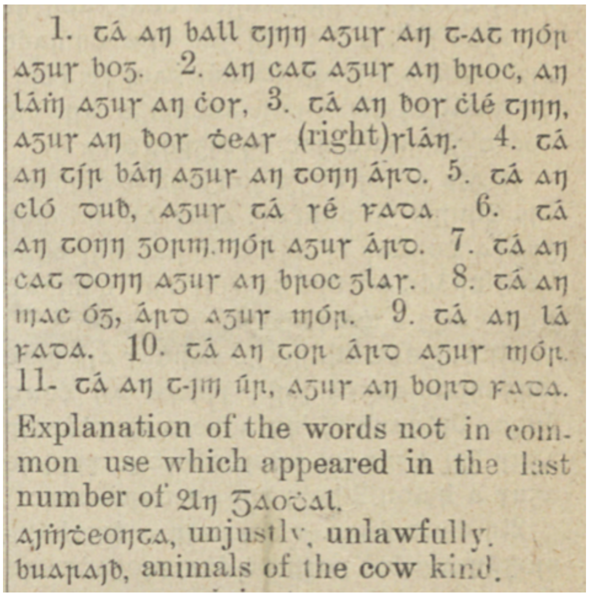
The only complete series of *An Gaodhal* spanning 1881 to 1898 survives in the James Hardiman Library at the University of Galway. This set was compiled, bound, and annotated by the Philadelphia-based scholar of Irish folklore and sean-nós song, Rev. Daniel J. Murphy, and forms part of his manuscript archive, which is also held in Galway (Ní Chonghaile, 2015). Since Rev. Murphy’s volumes of *An Gaodhal* were digitised in 2021 (University of Galway, 2021), the newspaper has been openly accessible as high resolution images via the University of Galway Library’s Digital Collections and Archives.[11] While the current interface provides searchable metadata, extending its functionality to include full-text searchability represents one of the ambitions of the present project, which aims to build a digitally enhanced edition of *An Gaodhal*.

As a monthly newspaper, *An Gaodhal* contained 12 numbers per volume. The corpus totals 147 issues from Vol. 1, No. 1, to Vol. 13, No. 3, and is complete and intact at 2,290 pages i.e. there are no missing pages. Most issues contain 16 pages; some contain 14, 12 or 8 pages. Page tears, ink spots, and blemishes are rare. Where such characteristics impair the legibility of text, human review relied on consulting the printed artefact or other extant samples of the relevant text.

The following list of key characteristics of the *An Gaodhal* corpus will help determine the relevance of the current project to the efforts of those seeking to apply OCR to other historical data:

- pages feature Irish mostly (381), English mostly (896), or both languages together (1,019);
- the use of two different typefaces throughout — Cló Gaelach and Roman — with infrequent changes of font and sometimes using Cló Gaelach for English content and Roman letters for Irish content (see Figures 2 and 3);
- the pre-standardised spelling of the Irish language in the late 19^{th} century;
- variations in spelling and vocabulary reflecting the three major dialects of Irish;
- variations in spelling reflecting the language aptitude of each contributor, many of whom were learners of the language or were gaining literacy in Irish for the first time;
- layout conventions reflecting the artisanal nature of the letterpress printing operation, which was small and domestic in scale and style, produced by the founder and editor Michael J. Logan entirely on a pro bono basis, and funded chiefly by subscriptions and advertisements.

4.1. Types, fonts, and marginalia



1. tá an ball tinn agus an t-at mór agus bog. 2. an cat agus an broc, an láṁ agus an ċos, 3. tá an ḃos ċlé tinn, agus an ḃos ḋeas (right)slán. 4. tá an tír bán agus an tonn árd. 5. tá an cló duḃ, agus tá sé fada 6. tá an tonn gorm.mór agus árd. 7. tá an cat donn agus an broc glas. 8. tá an mac óg, árd agus mór. 9. tá an lá fada. 10. tá an tor árd agus mór. 11. tá an t-im úr, agus an bord fada.

Explanation of the words not in common use which appeared in the last number of An Gaoḋal.

aiṁḋeonta, unjustly, unlawfully.

buaraiḃ, animals of the cow kind.

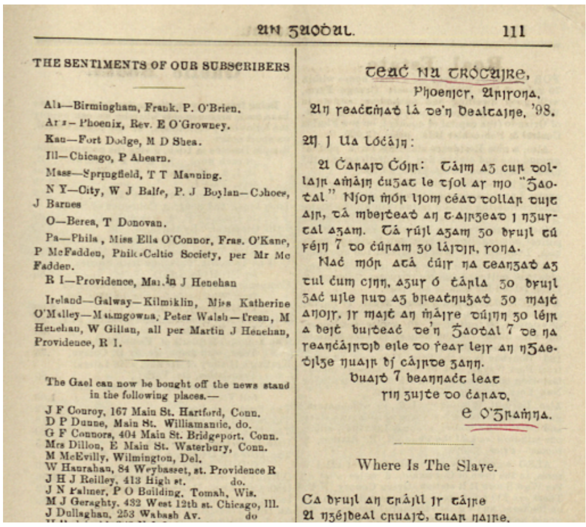
Figure 2: Example of mixed type usage in *An Gaodhal*.

The set of Cló Gaelach type used by Logan appears complete. A contemporary New York newspaper edited by Irish-born printer James Haltigan, *Celtic Monthly* (1879 – 1884), used a set of type that appears identical; however, the characters Ḃ, Ċ, Ḋ, Ḟ, Ġ, Ṁ, Ṗ, Ṡ, and Ṫ are applied variably therein (Knight, 2021). In lieu of dotted capital consonants, Haltigan and his colleagues sometimes rendered Ḃ, Ċ, and Ḋ as Bh, Ch, Dh, etc., a common substitution at this time and later where access to Cló Gaelach type was not guaranteed. To ensure that such nuances of contemporary typesetting and spelling conventions in a given printed artefact are preserved in the text extraction, the two new OCR models were trained to match a single Unicode character to each printed glyph; manually substituting Ḃ, Ċ, and Ḋ with Bh, Ch, and Dh, etc. was eschewed. Logan rarely adopted such substitutions and, in Irish-language texts, chose

1896, and Lyons (2021) on the Irish language revival, media and the transatlantic influence in 1857 – 1897.

[10] *An Gaodhal*, Vol. 9, No. 8 (January 1893): 236, accessible here: https://digital.library.universityofgalway.ie/p/ms/asset/16459

[11] https://digital.library.universityofgalway.ie/



AN GAOḊAL. 111

THE SENTIMENTS OF OUR SUBSCRIBERS

Ala—Birmingham, Frank. P. O'Brien.

Ar z—Phoenix, Rev. E O'Growney.

Kan—Fort Dodge, M D Shea.

Ill—Chicago, P Ahearn.

Mass—Springfield, T T Manning.

N Y—City, W J Balfe, P. J Boylan—Cohoes, J Barnes

O—Berea, T Donovan.

Pa—Phila , Miss Ella O'Connor, Fras. O'Kane, P McFadden, Phila Celtic Society, per Mr Mc Fadden.

R I—Providence, Martin J Henehan

Ireland—Galway—Kilmiklin, Miss Katherine O'Malley—Mamgowna, Peter Walsh—Crean, M Henehan, W Gillan, all per Martin J Henehan, Providence, R I.

The Gael can now be bought off the news stand in the following places.—

J F Conroy, 167 Main St. Hartford, Conn.
D P Dunne, Main St. Williamantic, do.
G F Connors, 404 Main St. Bridgeport. Conn.
Mrs Dillon, E Main St. Waterbury, Conn.
M McEvilly, Wilmington, Del.
W Hanrahan, 84 Weybasset, st. Providence R
J H J Reilley, 413 High st. do.
J N Palmer, P O Building, Tomah, Wis.
M J Geraghty, 432 West 12th st. Chicago, Ill.
J Dullaghan, 253 Wabash Av. do

TEAĊ NA TRÓCAIRE,
Phoenics, Arisona.
An seaċtṁaḋ lá de'n Bealtaine, '98.

M. I Ua Lóċáin:

A Ċaraid Ċóir: Táim ag cur dollair aṁáin ċugat le tsíol ar mo "Gaoṫal." Níor ṁór liom céad dollar duit air, dá mbeiṫeaḋ an t-airgead i ngustal agam. Tá súil agam go bfuil tú féin 7 do ċúram go láidir, sona.

Naċ mór atá ċúis na teangaḋ ag dul ċum cinn, agus ó tárla go bfuil gaċ uile rud ag breaṫnugaḋ go maiṫ anois, is maiṫ an ṁáise dúinn go léir a beiṫ buiṫeaċ de'n Gaoṫal 7 de na seanċáirdiḃ eile do ḟeas leis an nGaeḋilge nuair bí cáirde gann.

Buaiḋ 7 beannaċt leat
sin guiṫe do ċarad,
E O'Graṁna.

Where Is The Slave.

Ca bfuil an tráill is táire
A nGéibeal cruaiḋ, tuar naire.

Figure 3: Example of different Latin fonts and pica sizes in *An Gaodhal*.

to adhere to the relevant orthography, spelling some English words phonetically e.g. 'Nuaḋ Ġorc' for New York. In the present text extraction, the selected Unicode characters do not replicate exactly the design of the Cló Gaelach type such as Gaelchló[12] provides; rather, in deference to long-standing practice, Roman typeface characters — including those with diacritics (dots or accents) above the x-height or cap height e.g. ú or Ṁ — were chosen, thus ensuring interoperability between this dataset and others.

Printing errors are uncommon. Sometimes individual pieces of moveable type were placed in the printer's composing stick in the wrong order or upside-down, or supplies of particular letters e.g. a, á, e, é, ran short and were substituted with alternatives from either of the two orthographies. On occasion, insufficient ink or loose type rendered gaps. Corrections arising were tagged as 'supplied' or 'unclear' or 'gap' as appropriate to the word or line in question. Smaller pica sizes, which occurred only in the English-language fonts and most often in advertisements, proved challenging to the OCR software and thus prompted occasional manual text entry.

Handwritten marginalia corresponding to Rev. Murphy's handwriting occur on 495 of the 2,290 pages and were included in the OCR run. Appearing in black, blue, and red ink and in pencil, Rev. Murphy's annotations supply additional data including references to published books, journals, and newspapers; identify alternate song titles and associated song airs or melodies; and suggest corrections to the printed text content.

Abbreviations reflecting conventions of the period occur throughout, many of them serving to conserve space and type in printed matter, e.g. in English, 'Jas.' for 'James' and 'Patk.' for 'Patrick'. The names of American states are frequently abbreviated, with and without period marks and/or spacing, e.g. 'RI' and 'R. I.' for Rhode Island. In Irish, Logan frequently abbreviated 'agus' ('and') to the digit 7 in lieu of the Tironian symbol for the Latin 'et' (⁊). In correcting text extraction, human review reverted to the ampersand symbol (&) instead to avoid confusion with the digit 7.

5. OCR Workflow

The software selected for this process was READ-COOP's Transkribus (Kahle et al., 2017; Colutto et al., 2019), and the workflow included the following steps:

1. **Automating identification of predominantly Irish-language lines on pages.** This was done using Amazon's Textract software,[13] which could quickly and accurately produce token-detection and line segmentation regardless of language. The resulting OCR outputs were then categorised into Irish and non-Irish texts on a line-by-line basis by evaluating the dictionary-word accuracy of each line output. Pages scoring high as containing properly spelled English words were deemed 'non-Irish,' leaving a clear corpus of predominantly Irish-language pages to train an initial model. The team 'masked' English-language lines occurring in the pages of the selected corpus using overlaid opaque rectangles, enabling the creation of monolingual Irish-only page images.

2. **Training an OCR model for Irish-only pages.** From the masked Irish-only pages, the team selected 60 pages at random and, after excluding pages dominated by images or advertisements, a total of 57 proved viable for training. The team transcribed the texts on these pages manually and then used those transcriptions to create a model in Transkribus named *An Gaodhal Irish Model v. 1* for Cló Gaelach Irish-language detection (ID 50036), which incorporated 18,533 tokens.

3. **Training an OCR model on bilingual Irish-English pages.** The team selected 100 pages randomly from the entire collection, removing all masks to present fully bilingual texts. The language profile of each page determined which of the three selected OCR models ought to be applied. Pages predominantly in Irish were run through the Irish-only model (ID 50036); and pages predominantly

[12] https://www.gaelchlo.com/

[13] https://aws.amazon.com/textract

in English were run using *Transkribus Print M1* (ID 39995), which has been trained on over 5 million tokens and which also reflects the historical typographical conventions of the corpus. The resulting pages were then corrected manually, which provided the necessary content to train a bilingual OCR model. This bilingual model titled *An Gaodhal Irish / English Bilingual Model v. 1* (ID 51080) incorporated 54,406 input training tokens and achieved a character error rate (CER) close to 0 on the validation set.

4. **Correcting the outputs.** The team ran three different OCR models on the full 2,290 pages of the newspaper as appropriate to the language profile of each page: *Transkribus Print M1* on English-only or English-mostly pages; *An Gaodhal Irish Model v. 1* on the Irish-only or Irish-mostly pages; and *An Gaodhal Irish / English Bilingual Model v. 1* on bilingual pages. To date, half of these pages have been corrected by human review, including: all of the Irish-only pages; 41.9% of bilingual Irish-English pages; and 37.8% English-only pages (see Section 6.2 for more detail). With all 381 Irish-only or Irish-mostly pages corrected, a second Irish-only OCR model — *An Gaodhal Irish Model v. 2* (ID 61350) — was trained. It incorporated 164,015 words and achieved 1.4% CER on the validation set.

5. **Collecting supplementary page-level information.** The team reviewed and recorded key attributes of each page and presented the results of this review in the CSV file published together with the dataset. It lists: the presence of a table, advertisement, or image on each page; the language profile of the page — Irish, English, or bilingual; and the occurrence of verse (song or poem) or letters. This detail provides scope for further analysis of the content of the corpus (see Section 7 for the dataset description and reference).

6. OCR Post-Correction

6.1. Automatic correction

Whilst developing the bilingual OCR model, the team experimented with automatic OCR post-correction. The training set for OCR post-correction models included 103 pages from the 1 – 200 range; all of these pages had been manually corrected after the first application (as appropriate to the language profile of each page) of one of the project's chosen OCR models as outlined above. This dataset amounted to 9,994 lines of text and had 2.95% CER and 9.29% WER before manual correction. It was split into train and validation subsets with 0.9 : 0.1 ratio. The test set consisted of 235 lines from pages 10, 37 and 97 that were not used in the OCR model training. The test set CER and WER were 3.47% and 11.92% respectively.

The team decided to attempt fine-tuning state-of-the-art (SOTA) transformer models pre-trained for sequence-to-sequence tasks. In order to select the best transformer model for further experiments, we compared BART-base (Lewis et al., 2020), T5-base (Raffel et al., 2020), FLAN-T5-base (Chung et al., 2022), a BART-based English spellchecker (Guhr, 2023), and a T5-based spellchecker (Kundumani, 2022) by fine-tuning them with *An Gaodhal* data along with their default tokenisers. BART models performed significantly better than T5 models, as shown in Table 1.

Model	Test CER, %	Test WER, %
OCR output	3.47	11.92
BART-base	3.65	10.37
BART English spellchecker	3.40	10.50
T5-base	7.71	26.73
FLAN-T5-base	7.88	26.94
T5 English spellchecker	7.74	26.87

Table 1: Fine-tuning large language models on *An Gaodhal* data for OCR post-correction with default parameters.

The next step was to compare the performance of BART-base and BART-large models. Surprisingly, they demonstrated similar results: BART-base yielded 3.65% CER and 10.71% WER; and BART-large scores were 3.57% CER and 10.64% WER. As BART-large did not demonstrate a significant improvement compared to BART-base, the team proceeded with the smaller and less computationally-demanding BART-base model.

The team then measured how different tokenisers commonly used with transformer models[14] might influence performance. The standard tokeniser that comes with the BART-base model uses byte-level Byte-Pair Encoding, or BPE (Sennrich et al., 2016), and treats spaces like parts of the tokens. We trained three other tokenisers with slightly different architectures — SentencePiece (Kudo and Richardson, 2018), byte-level BPE, and character-level BPE — on bilingual Irish-English data from *An Gaodhal* and compared them to the standard BART tokeniser (see Table 2). BART-base performed best with our custom byte-level BPE tokeniser, achieving 3.33% CER and and 10.10% WER. This tokeniser was used in all subsequent experiments and is further referred to as 'custom tokeniser'.

[14] https://huggingface.co/docs/transformers/en/tokeniser_summary

Tokeniser	Test CER, %	Test WER, %
OCR output	3.47	11.92
Standard (BART-base)	3.65	10.71
Custom SentencePiece	3.63	10.37
Custom byte-level BPE	**3.33**	**10.10**
Custom char-level BPE	3.44	11.58

Table 2: The influence of different tokenisers on BART-base performance. The best result is marked in **bold**.

Finally, the team applied three data enhancement / fine-tuning techniques:

1. Masking. In large language models, it is common to mask 15% of tokens (Wettig et al., 2023) during pre-training to make the model more robust. The same strategy is recommended for BART fine-tuning.[15] Unfortunately, randomly masking 15% of words in our dataset at the pre-tokenisation stage did not yield better scores.

2. Balancing the dataset. The number of correct sentences in the training set for OCR post-correction outnumbered sentences containing OCR errors by a factor of 2.5. As such a class imbalance was likely to impede the efforts of a model to learn to correct errors, the team experimented with alternative ratios. We reduced the number of correct examples to a ratio of 1 : 1; and, in another set, to a slightly imbalanced ratio of 1 : 1.5, an adjustment that might mitigate the risk of over-correction. Though the model's performance improved in both settings, the difference between the 1 : 1 and 1 : 1.5 ratios was negligible.

3. Data augmentation. As the team aimed at training the model to correct very specific errors whilst also trying to avoid over-correction, it was decided to forgo introducing artificial noise. Instead, to augment the data, we elected to repeat every line in the dataset. However, the results revealed no improvement in the model's performance, either with 1 : 1 balancing or without.

The results are described in greater detail in Table 3. Analysing individual examples from the test set, we noted that models excel in correcting punctuation errors — such as an unnecessary or a missing space before/after a punctuation mark — or noise, usually in the form of dashes and square brackets. However, they are not as successful with incorrectly recognised letters, which is most likely due to the limited number of relevant examples in the training set.

All models were fine-tuned and tested with the help of the 'transformers' Python library (Wolf et al., 2020), and the best one is available on the HuggingFace model hub (Dereza, 2024) along with the corresponding dataset (Dereza et al., 2024).

6.2. Manual correction

The approach to correcting OCR output was curatorial, not editorial. As the newspaper was edited by the same individual from start to finish and printed under his guidance, there is a notable consistency of style throughout. Corrections were applied rarely and only then in the interests of ensuring discoverability. Non-standard forms of Irish-language spellings throughout prompted a strict adherence to the printed artefact as did printer's abbreviations — both conventional and idiosyncratic — that represent efforts to maximise space or optimise readability.

Punctuation and typographical conventions are generally preserved. However, some commas were inserted where printing rendered a period mark in the middle of a sentence; tilde marks ($\approx$) used in hyphenated compound words were replaced with a standard n-dash (–) to avoid confusion with the mathematical sign 'equal to' (=); and spaces were inserted on either side of m-dashes (—) to ensure that words on either side were recognised as separate entities. Some lines of text were justified from time to time but many more end with a word that is split between the end of that line and the start of the next, reflecting the physical restrictions of manual type-setting. In the printed artefact, the split is bridged by a n-dash (–). Excluding hyphenated compound words, we replaced such examples with the character ¬ (called a 'soft hyphen' or 'optional hyphen'). Such amendments aim to facilitate comprehension and deliver consistency for machine-reading tasks.

The bilingual Irish-English model (ID 51080) performed best when the content featured almost equal quantities of both languages and when the languages were confined to separate sections. Where the languages were intermixed in individual lines — in lists of translated Irish vocabulary or language instructional texts, for instance — the OCR output required more correction where the model failed to adjust to the rhythm of the orthographic exchanges on the page. Pages featuring a majority of English content required text entry for any Irish content therein where the English-only OCR model failed to render the Irish orthography. Likewise, pages featuring a majority of Irish content required text entry for any English content therein where the Irish-only OCR model failed to render the English orthography.

[15] https://huggingface.co/docs/transformers/model_doc/bart

Configuration	Train + valid data	Test CER, %	Test WER, %
OCR output	–	3.47	11.92
BART-base + standard tokeniser	9994 lines	3.65	**10.71**
BART-large + standard tokeniser	9994 lines	3.57	**10.64**
BART-base + custom tokeniser	9994 lines	**3.33**	**10.10**
BART spellchecker + custom tokeniser	9994 lines	**3.40**	**10.24**
BART-base + custom tokeniser + masking	9994 lines	3.60	**10.91**
BART-base + custom tokeniser + data balanced 50:50	5734 lines	**3.29**	**9.83**
BART-base + custom tokeniser + data balanced 40:60	7154 lines	**3.29**	**9.83**
BART-base + custom tokeniser + data augmented x2	19988 lines	**3.39**	**10.24**
BART-base + custom tokeniser + data augmented x2, balanced 50:50	11468 lines	**3.27**	**10.17**

Table 3: Comparison of different BART fine-tuning configurations. Improvements in CER / WER on the test set are marked in **bold**, and the best result is **underlined**.

Where OCR failed to render complete lines or word boxes, these were entered manually. Lines were sometimes joined or split to maximise comprehensibility of the extracted text. Corrections were provided at word-level, not simply at line-level, to enable future application of language-based technologies.

As is common in OCR workflows, layout detection was important to overall accuracy, especially given that columns and paragraph structures were used by the printers throughout. To yield workable baseline recognition, print block detection and layout analysis models offered by Transkribus were applied — at default settings — consecutively to each page. The occurrence of two columns on most pages, tables, advertisements, images, marginalia, and fine print demanded careful review of the page layout and sometimes required manual treatment including adjusting baselines and box boundaries and hand-drawing baselines for vertical text and marginalia.

7. Output

The work described above resulted in the machine-readable full text of *An Gaodhal* published on the NYU UltraViolet platform (Ní Chonghaile et al., 2023). The data constitute direct exports from Transkribus of the resulting full text. The files are presented in two forms:

1. Alto-format XML files that provide bounding box regions for text locations (at the individual token level) of separately tokenised pages,
2. Page-format XML files, which are comparable to Alto files but use a specific output format for Transkribus software.

XML files are internally self-describing, with tags providing names of fields. 'Page' Transkribus output format files are organised on a per-page basis into regions (<TextRegion>) or tables (<TableRegion>), lines (<TextLines>) or table cells (<TableCell>) respectively, and words (<Word>). Regions are also labeled according to a structure type: paragraph (<TextRegion type='paragraph'>), page-number (<TextRegion type='page-number'>), or marginalia (<TextRegion type='marginalia'>). These distinguish between the standard printed newspaper text, a page number printed on the page, and handwritten marginalia added to the original printed artefact.

Each structural element maps to the image uploaded to the software, reading each of the newspaper's two columns left to right from top to bottom. Exceptions arise where the usual layout deviates according to the printer's prerogative; for instance, when a reader's eye moves at intervals over and back between the two columns. In such rare cases, human review prompted the re-ordering of the sequence to ensure the extracted text output was as logical and comprehensible as the experience of reading the printed artefact.

Each region, line, and word has a unique identifier derived from its logical sequence on the page. Thus, for example, word id 'r5l1w2' refers to region 5, line 1, word 2. Tables, table cells, and words conform to the same style of sequencing e.g. region id 'tbl_4_4' refers to a table appearing between Regions 3 and 4 of standard text areas, and the relevant word id entries appear per line and word (left to right) as 'r_4_1_1' and 'r_4_1_2'.

Additional identifiers indicating separators (<Separator>) are retained in the data. The separator ID numbers do not conform to the sequence of identifiers mapping all other page elements; rather, they retain the identifiers generated automatically by the initial layout analysis. Hence, they appear somewhat random — two consecutive separators might appear as 'r_25' and 'r_39' — and are typically grouped together at the end of the page metadata. Each separator corresponds to a decorative hairline rule or border demarcating different elements of the printed page, separating articles or advertisements from each other. Such decorative elements aid the reader's navigation of a printed page. In a digital environment, an equivalent distinction is provided by the structural tags applied during text extraction. As such, separators were deemed surplus to the requirements of text extraction. In addition, such was the quantity of separators throughout, time did not allow for the re-sequencing of each individual separator between different text regions as they appear on each page.

Bounding coordinates for polygons and locations of points making up text baselines are oriented to an origin point (0,0) at the top left of the page, mapping each element to the image in question. X,Y coordinates are given as pairs in the form x1,y1; x2,y2; etc.

ALTO output format files follow the XML stylesheet maintained by the Library of Congress.[16] These files follow a similar region, line, string format, with the token provided at <string CONTENT>.

The accompanying CSV[17] provides additional metadata on a per-page basis that were recorded in the course of page layout review. Page metadata appear in rows with columns distinguishing between the following elements: page filename; the language profile of the page — Gaeilge (Irish), English, or Bilingual; presence of skew or tight gutters; and whether or not a page contains marginalia, images, advertisements, verse, or letters. Variables include:

- pageFilename: XML OCR output to which row data refer
- skew_gutter_fallaway: Yes/No on presence of a skew, gutter, or fallaway on digitised page that might affect OCR quality
- hasTable: Yes/No on presence of a table or table-like arrangement of tokens on page (includes list and list-like structures)
- language: Gaeilge/English/Mix, predominant language on page
- isCover: Yes/No on whether this page is the issue start (i.e. cover) page
- hasMarginalia: Yes/No on whether handwritten margin notes are present
- hasSong_Poem: Yes/No on whether a song or poem, or part thereof, is present
- hasAdvert: Yes/No on whether an advertisement is present
- hasLetter: Yes/No on whether a letter, or part thereof, is present
- hasImage: Yes/No on whether an image is present

8. Future Work

The team has begun working on Named Entity Recognition (NER) for Irish toward automatically extracting references to people, events, locations, dates, creative works, and more from the text. For this purpose, a dataset of 11,000 words was labeled manually according to IOBES annotation scheme to train / fine-tune a deep learning model for historical Irish NER in future. To the best of our knowledge, this attempt is the first of its kind for the Irish language. As for the English-language content from *An Gaodhal*, we applied NER to it separately using *en_core_web* models trained on large English-language corpora available through Python NLP framework spaCy (2023).

The team is also identifying corpora suitable for future applications of these new OCR and NER tools, e.g. the bilingual Irish-English newspaper *An Stoc*, 1917-1931 (University of Galway, 2022).

9. Conclusion

The project has presented its team with a unique set of challenges, some of which have been explored previously by only a handful of initiatives.

All project outputs will be made publicly accessible and available for further application in the field of computational linguistics. Creating an open interface enabling searches of the bilingual content of *An Gaodhal* will reveal to a wider public the vitality of Irish language practice in a diasporic context and reflect its co-existence alongside English. This new resource will enable historians to better contextualise the multilingual heritage of the Irish diaspora. The specificity of the newspaper's content and readership will be a particular boon to genealogists.

The OCR, OCR post-correction and NER tools produced by this project represent welcome additions to the digital tool-kit serving the Irish language into the future. Finally, the methodologies described here may come to inform and so serve other under-resourced and endangered languages worldwide.

10. Acknowledgements

This project has been supported to date by the Robert D. L. Gardiner Foundation, the Irish Institute of New York, New York University Glucksman Ireland House, and the University of Galway. We express our special gratitude to the James Hardiman Library at University of Galway for providing high-resolution images of *An Gaodhal*.

[16] Version 4.4 is the most current at the time of submission: https://www.loc.gov/standards/alto/v4/alto-4-4.xsd

[17] https://ultraviolet.library.nyu.edu/records/5ya5n-mc504/files/AnGaodhal_pageMetadata.csv

11. Authors' Contributions

Oksana Dereza focused on the NLP technologies, provided analysis of related work on OCR and NER, carried out computer-assisted OCR post-correction and NER experiments, and lead the writing of this paper.

Deirdre Ní Chonghaile provided historical and linguistic domain expertise, reviewed each corpus page toward generating the metadata CSV, conducted transcription work toward training OCR models on Transkribus, ran OCR, and performed manual OCR correction.

Nicholas Wolf acted as project PI, provided historical and linguistic domain expertise, conducted transcription work toward training OCR models on Transkribus and trained those models, reviewed metadata CSV, and prepared a NER training dataset that was reviewed by Deirdre.

All authors contributed to the project design and to the final manuscript.

12. Bibliographical References

Haithem Afli, Zhengwei Qiu, Andy Way, and Páraic Sheridan. 2016. Using SMT for OCR error correction of historical texts. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, pages 962–966. European Language Resources Association (ELRA).

Clément Besnier and William Mattingly. 2021. Named-entity dataset for medieval Latin, Middle High German and Old Norse. *Journal of Open Humanities Data*, 7(0):23.

Tobias Blanke, Michael Bryant, and Mark Hedges. 2012. Open source optical character recognition for historical research. *Journal of Documentation*, 68(5):659–683.

Emanuela Boros, Elvys Linhares Pontes, Luis Adrián Cabrera-Diego, Ahmed Hamdi, José G Moreno, Nicolas Sidère, and Antoine Doucet. 2020. Robust named entity recognition and linking on historical multilingual documents. In *Conference and Labs of the Evaluation Forum (CLEF 2020)*, volume 2696 of *CEUR Workshop Proceedings*, pages 1–17. CEUR-WS.

Syed Saqib Bukhari, Ahmad Kadi, Mohammad Ayman Jouneh, Fahim Mahmood Mir, and Andreas Dengel. 2017. anyOCR: An open-source ocr system for historical archives. In *Proceedings of the 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 305–310. IEEE.

Carlotta Capurro, Vera Provatorova, and Evangelos Kanoulas. 2023. Experimenting with training a neural network in Transkribus to recognise text in a multilingual and multi-authored manuscript collection. *Heritage*, 6(12):7482–7494.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv:2210.11416*.

Sebastian Colutto, Philip Kahle, Hackl Guenter, and Günter Mühlberger. 2019. Transkribus: A platform for automated text recognition and searching of historical documents. In *Proceedings of the 15th International Conference on eScience (eScience)*, pages 463–466. IEEE.

Dana Dannélls and Simon Persson. 2020. Supervised OCR post-correction of historical Swedish texts: What role does the OCR system play? In *Proceedings of the Digital Humanities in the Nordic Countries, 5th Conference, Riga, Latvia, October 21-23, 2020*, volume 2612 of *CEUR Workshop Proceedings*, pages 24–37. CEUR-WS.

Ghaith Dekhili and Fatiha Sadat. 2020. Hybrid statistical and attentive deep neural approach for named entity recognition in historical newspapers. In *Working Notes of CLEF 2020 – Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.

İshak Dölek and Atakan Kurt. 2022. A deep learning model for Ottoman OCR. *Concurrency and Computation: Practice and Experience*, 34(20):e6937.

Rui Dong and David A Smith. 2018. Multi-input attention for unsupervised OCR correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2363–2372. Association for Computational Linguistics.

Senka Drobac. 2020. *OCR and post-correction of historical newspapers and journals*. Ph.D. thesis, University of Helsinki.

Senka Drobac, Pekka Kauppinen, and Krister Lindén. 2017. OCR and post-correction of historical Finnish texts. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 70–76. Association for Computational Linguistics.

Quan Duong, Mika Hämäläinen, and Simon Hengchen. 2021. An unsupervised method for OCR post-correction and spelling normalisation for Finnish. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 240–248. Linköping University Electronic Press, Sweden.

Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. 2023. Named entity recognition and classification in historical documents: A survey. *ACM Computing Surveys*, 56(2).

Maud Ehrmann, Matteo Romanello, Alex Flückiger, and Simon Clematide. 2020. Extended overview of CLEF HIPE 2020: Named entity processing on historical newspapers. In *CLEF 2020 Working Notes. Conference and Labs of the Evaluation Forum*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.

Maud Ehrmann, Matteo Romanello, Sven Najem-Meyer, Antoine Doucet, and Simon Clematide. 2022. Extended overview of HIPE-2022: Named entity recognition and linking in multilingual historical documents. In *Proceedings of the Working Notes of CLEF 2022 – Conference and Labs of the Evaluation Forum*, volume 3180 of *CEUR Workshop Proceedings*. CEUR-WS.

Florian Fink, Klaus U Schulz, and Uwe Springmann. 2017. Profiling of OCR'ed historical texts revisited. In *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage*, DATeCH 2017, page 61–66. Association for Computing Machinery.

Lenz Furrer and Martin Volk. 2011. Reducing OCR errors in Gothic-script documents. In *Proceedings of the Workshop on Language Technologies for Digital Humanities and Cultural Heritage*, pages 97–103. Association for Computational Linguistics.

Government of Ireland. 2022. Digital plan for the Irish language. Speech and language technologies 2023-2027. Accessed: 29 March 2024.

Ivan Gruber, Marek Hrúz, Pavel Ircing, Petr Neduchal, Tomáš Zítka, Miroslav Hlaváč, Zbyněk Zajíc, Jan Švec, and Martin Bulín. 2021. OCR improvements for images of multi-page historical documents. In *International Conference on Speech and Computer (SPECOM 2021)*, volume 12997 of *Lecture Notes in Computer Science*, pages 226–237. Springer.

Helena Hubková. 2019. Named-entity recognition in Czech historical texts: Using a CNN-BiLSTM neural network model. Master's Thesis in Language Technology, Uppsala University.

Helena Hubková, Pavel Král, and Eva Pettersson. 2020. Czech historical named entity corpus v. 1.0. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4458–4465. European Language Resources Association.

Philip Kahle, Sebastian Colutto, Günter Hackl, and Günter Mühlberger. 2017. Transkribus – a service platform for transcription, recognition and retrieval of historical documents. In *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 4, pages 19–24. IEEE.

Kimmo Kettunen, Mika Koistinen, and Jukka Kervinen. 2020. Ground truth OCR sample data of Finnish historical newspapers and journals in data improvement validation of a re-ocring process. *LIBER Quarterly: The Journal of the Association of European Research Libraries*, 30(1):1–20.

Matthew Knight. 2021. *"Our Gaelic Department": The Irish-Language Column in the New York Irish-American, 1857-1896*. Ph.D. thesis, Harvard University Graduate School of Arts and Sciences.

Mika Koistinen, Kimmo Kettunen, and Jukka Kervinen. 2020. How to improve optical character recognition of historical Finnish newspapers using open source Tesseract OCR engine – final notes on development and evaluation. *Human Language Technology. Challenges for Computer Science and Linguistics*, pages 17–30.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Association for Computational Linguistics.

Fiona Lyons. 2021. *Thall is abhus: Irish language revival, media and the transatlantic influence 1857-1897*. Ph.D. thesis, University College Dublin.

Lijun Lyu, Maria Koutraki, Martin Krickl, and Besnik Fetahu. 2021. Neural OCR post-hoc correction of historical corpora. *Transactions of the Association for Computational Linguistics*, 9:479–493.

Hsing-Yuan Ma, Hen-Hsen Huang, and Chao-Lin Liu. 2024. Reading between the lines: Image-based order detection in OCR for Chinese historical documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38:21, pages 23808–23810. The Association for the Advancement of Artificial Intelligence.

Jiří Martínek, Ladislav Lenc, and Pavel Král. 2020. Building an efficient OCR system for historical documents with little training data. *Neural Computing and Applications*, 32:17209–17227.

Dermot McGuinne. 1992. *Irish type design: A history of printing types in the Irish character*. Art and Architecture Series. Irish Academic Press.

Christopher Moseley, editor. 2010. *Atlas of the World's Languages in Danger*, 3rd edition. Memory of Peoples. UNESCO Publishing, Paris.

Clemens Neudecker. 2016. An open corpus for named entity recognition in historic newspapers. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, pages 4348–4352. European Language Resources Association (ELRA).

Deirdre Ní Chonghaile. 2015. "Sagart gan iomrádh": An tAthair Domhnall Ó Morchadha (1858-1935) agus amhráin Philadelphia. In Máirín Nic Eoin, Ríona Nic Congáil, Pádraig Ó Liatháin, Meidhbhín Ní Úrdail, and Regina Uí Chollatáin, editors, *Litríocht na Gaeilge ar fud an Domhain*, pages 191–214. *Leabhair*COMHAR, Baile Átha Cliath.

Vera Provatorova, Svitlana Vakulenko, Evangelos Kanoulas, Koen Dercksen, and Johannes M. van Hulst. 2020. Named entity recognition and linking on historical newspapers: UvA. ILPS & REL at CLEF HIPE 2020. In *CLEF 2020: CLEF 2020 Working Notes: Working Notes of CLEF 2020 – Conference and Labs of the Evaluation Forum*, volume 2696 of *CEUR Workshop Proceedings*, pages 1–8, Thessaloniki, Greece. CEUR-WS.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Christian Reul. 2020. *An Intelligent Semi-Automatic Workflow for Optical Character Recognition of Historical Printings*. Ph.D. thesis, Bayerische Julius-Maximilians-Universität Würzburg (Germany).

Christian Reul, Christoph Wick, Maximilian Nöth, Andreas Büttner, Maximilian Wehner, and Uwe Springmann. 2021. Mixed model OCR training on historical Latin script for out-of-the-box recognition and finetuning. In *The 6th International Workshop on Historical Document Imaging and Processing*, HIP'21, pages 7–12. Association for Computing Machinery.

Martin Reynaert. 2008. Non-interactive OCR post-correction for giga-scale digitization projects. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 617–630. Springer.

Martin Reynaert. 2016. OCR post-correction evaluation of Early Dutch books online – revisited. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, pages 967–974. European Language Resources Association (ELRA).

Caitlin Richter, Matthew Wickes, Deniz Beser, and Mitch Marcus. 2018. Low-resource post processing of noisy OCR output for historical corpus digitisation. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).

Shruti Rijhwani, Antonios Anastasopoulos, and Graham Neubig. 2020. OCR Post Correction for Endangered Language Texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5931–5942. Association for Computational Linguistics.

Shruti Rijhwani, Daisy Rosenblum, Antonios Anastasopoulos, and Graham Neubig. 2021. Lexically aware semi-supervised learning for OCR post-correction. *Transactions of the Association for Computational Linguistics*, 9:1285–1302.

Jeff Rusten. 2020. Training a multilingual model in Transkribus. Transkribus Blog. Accessed: 29 March 2024.

Sarah Schulz and Jonas Kuhn. 2017. Multi-modular domain-tailored OCR post-correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2726, Copenhagen, Denmark. Association for Computational Linguistics.

Stefan Schweter and Johannes Baiter. 2019. Towards robust named entity recognition for historic German. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 96–103. Association for Computational Linguistics.

Stefan Schweter, Luisa März, Katharina Schmid, and Erion Çano. 2022. hmBERT: Historical multilingual language models for named entity recognition. In *Proceedings of the Working Notes of CLEF 2022 – Conference and Labs of the Evaluation Forum*, volume 3180 of *CEUR Workshop Proceedings*. CEUR-WS.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Prawaal Sharma, Poonam Goyal, Vidisha Sharma, and Navneet Goyal. 2024. VOLTAGE: A versatile contrastive learning based OCR methodology for ultra low-resource scripts through auto glyph feature extraction. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–899. Association for Computational Linguistics.

Richard Sharpe and Micheál Hoyne. 2020. *Clóliosta: Printing in the Irish Language, 1571-1871*. Dublin Institute for Advanced Studies, Dublin.

David A. Smith and Ryan Cordell. 2018. A research agenda for historical and multilingual optical character recognition. Accessed: 29 March 2024.

Elizabeth Soper, Stanley Fujimoto, and Yen-Yun Yu. 2021. BART for post-correction of OCR newspaper text. In *Proceedings of the 7th Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 284–290. Association for Computational Linguistics.

Uwe Springmann, Christian Reul, Stefanie Dipper, and Johannes Baiter. 2018. Ground truth for training OCR engines on historical documents in German Fraktur and Early Modern Latin. *Journal for Language Technology and Computational Linguistics*, 33(1):97–114.

Omri Suissa, Maayan Zhitomirsky-Geffet, and Avshalom Elmalech. 2022. Toward a period-specific optimized neural network for OCR error correction of historical Hebrew texts. *ACM Journal on Computing and Cultural Heritage (JOCCH)*, 15(2):1–20.

Marcella Tambuscio and Tara Lee Andrews. 2021. Geolocation and named entity recognition in ancient texts: A case study about Ghewond's Armenian history. In *Proceedings of the Conference on Computational Humanities Research 2021*, volume 2989 of *CEUR Workshop Proceedings*, pages 136–148. CEUR-WS.

Konstantin Todorov and Giovanni Colavizza. 2020. Transfer learning for named entity recognition in historical corpora. In *Working Notes of CLEF 2020 – Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.

Solenn Tual, Nathalie Abadie, Joseph Chazalon, Bertrand Duménieu, and Edwin Carlinet. 2023. A benchmark of nested named entity recognition approaches in historical structured documents. In *Document Analysis and Recognition – ICDAR 2023*, pages 115–131. Springer Nature Switzerland.

Fionnuala Uí Fhlannagáin. 1990. *Mícheál Ó Lócháin agus An Gaodhal*. An Clóchomhar Tta., Baile Átha Cliath.

Thorsten Vobl, Annette Gotscharek, Uli Reffle, Christoph Ringlstetter, and Klaus U. Schulz. 2014. PoCoTo – an open source system for efficient interactive postcorrection of OCRed historical texts. In *Proceedings of the 1st International Conference on Digital Access to Textual Cultural Heritage (DATeCH 2014), Madrid, Spain, May 19-20, 2014)*, pages 57–61. Association for Computing Machinery.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Tariq Yousef, Chiara Palladino, and Stefan Jänicke. 2023. Transformer-based named entity recognition for Ancient Greek. In *Digital Humanities 2023: Book of Abstracts*, pages 194–195. Zenodo.

13. Language Resource References

Acadamh Ríoga na hÉireann. 2017. *Corpas Stairiúil na Gaeilge 1600-1926*. Royal Irish Academy. Accessed: 29 March 2024.

Dereza, Oksana. 2024. *BART-base fine-tuned for OCR post-correction of historical bilingual Irish-English data*. HuggingFace. Accessed: 29 March 2024.

Dereza, Oksana and Ní Chonghaile, Deirdre and Wolf, Nicholas. 2024. *Historical bilingual Irish-English dataset for OCR post-correction*. HuggingFace. Accessed: 29 March 2024.

Facebook. 2022. *BART-base*. HuggingFace. Accessed: 29 March 2024.

Farrell, Gerard. 2023. *Irish, Gaelic and Roman type (Seanchló agus Cló Rómhánach) v.3*. Transkribus. Accessed: 29 March 2024.

Dublin Institute for Advanced Studies. 2019. Irish script on screen. Website. Accessed: 29 March 2024.

Google. 2023a. *FLAN-T5-base*. HuggingFace. Accessed: 29 March 2024.

Google. 2023b. *T5-base*. HuggingFace. Accessed: 29 March 2024.

Guhr, Oliver. 2023. *Spelling Correction English Base*. HuggingFace. Accessed: 29 March 2024.

Kundumani, Bhuvana. 2022. *T5 Base Spellchecker*. HuggingFace. Accessed: 29 March 2024.

Ní Chonghaile, Deirdre and Dereza, Oksana and Wolf, Nicholas. 2023. *An Gaodhal Newspaper (1881-1898): Full-Text OCR Output Files (Version 1)*. New York University. Accessed: 29 March 2024.

Scannell, Kevin and Regan, Jim and Damazyn, Kevin. 2020. *Tesseract Irish Uncial Training Data*. GitHub. Accessed: 29 March 2024.

Schweter, Stefan. 2020. *Europeana BERT and ELECTRA models (1.0.0)*. Zenodo. Accessed: 29 March 2024.

spaCy. 2023. English language models. Website. Accessed: 29 March 2024.

Transkribus Team. 2021. *Transkribus print M1*. Transkribus. Accessed: 29 March 2024.

University of Galway. 2021. *An Gaodhal Newspaper*. University of Galway. Accessed: 29 March 2024.

University of Galway. 2022. *An Stoc Newspaper*. University of Galway. Accessed: 29 March 2024.

Introducing PaVeDa – Pavia Verbs Database: Valency Patterns and Pattern Comparison in Ancient Indo-European Languages

Silvia Luraghi, Alessio Palmero Aprosio, Chiara Zanchi, Martina Giuliani
University of Pavia, Bruno Kessler Foundation Trento, University of Pavia, Universities of Pavia and Bergamo
luraghi@unipv.it, aprosio@fbk.eu, chiara.zanchi@unipv.it, martina.giuliani@unibg.it

Abstract

The paper introduces PaVeDa (Pavia Verbs Database), a resource that builds on the ValPaL database of verbs' valency patterns and alternations by adding a number of ancient languages (completely absent from ValPaL) and a number of new features that enable direct comparison, both diachronic and synchronic. For each verb, ValPaL contains the basic frame and ideally all possible valency alternations allowed by the verb (e.g. passive, causative, reflexive etc.). In order to enable comparison among alternations, an additional level has been added, the alternation class, that overcomes the issue of comparing language specific alternations which were added by individual contributors of ValPaL. The ValPaL had as its main aim typological comparison, and data collection was variously carried out using questionnaires, secondary sources and largely drawing on native speaker intuition by contributors. Working with ancient languages entails a methodological change, as the data is extracted from corpora. This has led to re-thinking the notion of valency as a usage-based feature of verbs and to planning future addition of corpus data to modern languages in the database. It further shows the impact of ancient languages on theoretical reflection.

Keywords: verbal valency, valency patterns, alternations.

1. Introduction

In this paper we introduce a newly created resource, PaVeDa (the Pavia Verbs Database, https://paveda.unipv.it/), which expands on an existing one, the ValPaL database. The latter is a typological database, intended to document valency patterns and alternations in a variety of languages of different areal and genealogical affiliation, in which data from each language can be visualized in isolation. Our new database builds on the existing resource to include ancient Indo-European languages and adds features that allow visualizing direct comparison among languages (Zanchi, Luraghi and Combei 2022).

The paper is organized as follows. In Section 2 we briefly introduce the original ValPaL database and focus on some issues in data collection and presentation that affect cross-linguistic comparability. In Section 3 we describe the new features of PaVeDa and show its possible uses for synchronic and diachronic language comparison. In Section 4 we outline our plans for further extension of PaVeDa. Section 5 contains the conclusion.

2. Background: the ValPaL database

The ValPaL (Valency Patterns Leipzig Online Database) available at https://valpal.info/ is one of the main results of the Leipzig Valency Classes Project, carried out from 2009 to 2013, aimed at a large-scale cross-linguistic comparison of valency classes. The ValPaL project follows up on Levin's (1993) intuitions of providing a semantic classification of verbs based on their syntactic behavior. Valency classes are conceived as groups of verbs sharing morphosyntactic properties, i.e. coding patterns and valency alternations.

The ValPaL database stores information regarding the basic valency patterns and alternations of a selection of verb meanings for 36 languages belonging to 23 language families. The ValPaL verb selection singled out 80 core verb meanings, based on two criteria: a) representativeness of the entire verbal lexicon; b) known instantiations of distinctive grammatical behavior according to previous studies. These verb meanings denote a variety of events with different numbers of participants. They include two-place changes-of-state verbs (e.g. BREAK, KILL), three-place verbs of transfer (e.g. GIVE, BRING) and cognitive transfer (e.g. TELL, TEACH), perception verbs (e.g. SEE, SMELL), verbs of cognitions (e.g. THINK), emotions (e.g. LIKE) and bodily sensations (e.g. BE HUNGRY), activities (e.g. RUN, LAUGH), and weather verbs (e.g. RAIN), which are cross-linguistically zero-place verbs. Each verb meaning is paired with the semantically most fitting basic verb in each project language. Additional verb meanings were occasionally included for specific languages, up to the total of 162 verb meanings currently represented in the database. Only the 80 core verb meanings are represented in the database for each project language, resulting in a partial coverage of the newly added meanings. For example, the basic verb for WINK is stored for three languages, whereas the core meaning BLINK is covered by 35 languages; the meaning ASSASSINATE is available only for Italian.

The basic valency pattern and possible valency alternations available as 'coding frames' are stored along with each verb (cf. Section 2.1). Alternations are classified as 'coded' if morphologically marked on the verb, or as 'uncoded' if unmarked. As its original aim is not diachronic analysis, the ValPaL does not include data from ancient languages.

The ValPaL database has paved the way for the subsequent creation of similar typological databases, e.g. BivalTyp database (which can be found at https://www.bivaltyp.info/), which stores bivalent verbs and their encoding frames for 124 languages (Say 2014). Partly inspired by ValPaL is also the Multilingual Verb Valence Lexicon, which offers verb valency information in a uniform format for four languages: Norwegian, Spanish, Ga and Bulgarian (Hellan et al. 2014). The ValPaL database is

LT4HALA 2024@LREC-COLING-2024, pages 79–88
25 May, 2024.

commonly considered a valuable tool for synchronic cross-linguistic investigations of valency patterns (Malchukov and Comrie 2015), and several studies that relied on its data have achieved important results (e.g. Aldai and Wichmann 2018). In this framework, the fact that no diachronic research is supported by the data stored in ValPaL is an important shortcoming, and ultimately also affects typological comparison. Moreover, even though single valency-related phenomena and certain argument structure constructions are well-studied topics for some ancient Indo-European languages, even these languages generally lack comprehensive overviews of their valency classes and alternations. A partial fill of this gap can be found in the valency lexica automatically induced from treebanks, i.e. morpho-syntactically annotated corpora in which dependency structures are stored as syntactic trees. Currently, such valency lexica are available for a limited set of ancient languages, notably Latin and Ancient Greek (McGillivray et al. 2009, McGillivray and Passarotti 2015, McGillivray and Vatri 2015, Passarotti et al. 2016, Zanchi et al. 2018, Zanchi 2021), and for the ancient languages included in the PROIEL project (available at http://dev.syntacticus.org/proiel.html), i.e. Latin, Ancient Greek, Old Russian, Old Church Slavic, Gothic, Old English, Classical Armenian and Old French. The new PROIEL treebank browser, Syntacticus, allows for visualization of the so-called “valency table”, in which argument structure constructions with relative frequencies are given for verbs. These valency tables, too, are automatically generated from the syntactic annotation in the treebanks of the PROIEL project. A valency lexicon of Classical Armenian is currently under construction at the University of Wüzburg in the framework of the project CAVAL – The Classical Armenian Valency Lexicon (see https://www.phil.uni-wuerzburg.de/en/vgsp/research/projects/). As discussed at length by Zanchi et al. (2018) and Zanchi (2021), valency lexica of this type are useful resources if employed with caution: they reflect the classification system for arguments and adjuncts indicated in the annotation guidelines, may contain annotation errors inherited from treebanks, and do not account for null referential arguments, widespread in ancient Indo-European languages and not annotated in treebanks (see Luraghi 2003, Keydana and Luraghi 2012, Haug 2012, Sausa and Zanchi 2015).

Thus, a valency database compiled by humans and storing valency frames of verbs from ancient languages is certainly a *desideratum*. Notably, some work in this direction has also been done in the framework of the LiLa project (see https://lila-erc.eu/#page-top), whereby valency frames are added to the verbal synsets contained in the Latin WordNet (Mambrini et al. 2021). Similar endeavors with Sanskrit and Ancient Greek are described in Biagetti et al. (2023a, 2023b).

2.1 The data available in ValPaL

For each meaning stored in ValPaL one can find and visualize data related to individual languages, the geographical distribution, and the list of alternations available across languages, as shown in Figure 1.

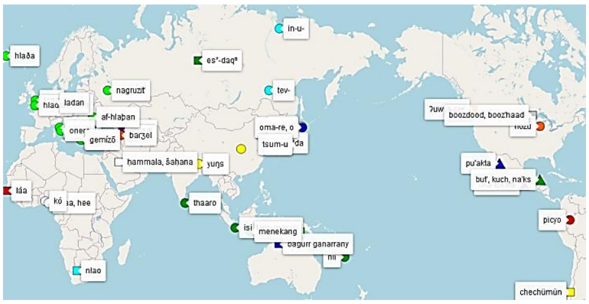
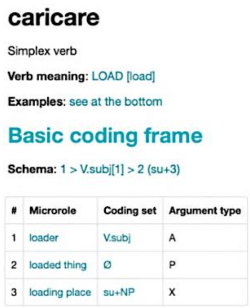


Figure 1. Instantiations of the meaning LOAD

All verb meanings stored in ValPaL are cross-referenced with Concepticon, a resource that links concept labels from different concept lists to concept sets, which are given a unique identifier, a unique label and a human-readable definition (see https://concepticon.clld.org). Languages stored in ValPaL are paired with their Glottocodes, i.e. unique and stable identifiers that allow ValPaL to be cross-referenced with Glottolog (see https://glottolog.org/glottolog/language).
All coding frames and alternations are illustrated with examples including grammatical glosses and translations; verb-specific microroles and information about word order and argument type are also featured in the coding frame. As an example, let us consider the Italian verb *caricare* ‘load’ in Figure 2.



caricare

Simplex verb

Verb meaning: LOAD [load]

Examples: see at the bottom

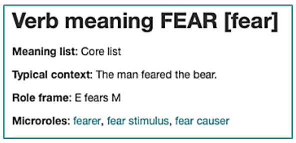
Basic coding frame

Schema: 1 > V.subj[1] > 2 (su+3)

#	Microrole	Coding set	Argument type
1	loader	V.subj	A
2	loaded thing	Ø	P
3	loading place	su+NP	X

Figure 2: The coding frame of Italian *caricare*

In the coding frame the symbol > indicates word order, [...] indicates agreement and (...) optionality. The coding set refers to the morphological marking of the arguments, while possible argument types are A (transitive verb subject), P (direct object), S (intransitive verb subject), I (instrument), L (locative) and X (other).



Verb meaning FEAR [fear]

Meaning list: Core list

Typical context: The man feared the bear.

Role frame: E fears M

Microroles: fearer, fear stimulus, fear causer

Figure 3: role frame and microroles for FEAR

Along with each verb meaning, a list of microroles is provided. Figure 3 shows the microroles associated with FEAR. From Figure 2, one can see that each microrole is assigned a number, for example #1 indicates the “loader”, #2 “the loading thing” and #3 ”the loading place”. This numbering is crucial to interpreting basic and derived coding frames and to understanding how microroles are mapped to different basic and alternating argument structure constructions. For example, as shown in Figure 2, "1 > V.subj[1] > 2 (su+3)" is the basic coding frame of Italian LOAD *caricare*. The same assignment of numbers to microroles is kept in the derived coding frame of the passive alternation: "2 > passV'.subj[2] (su+3) (daParteDi+1)", from which one understands that microrole #2 "loaded thing" is passivized and microrole #1 "loader" becomes non obligatory, as expected for passive agents.

In the ValPaL database there are a number of inconsistencies concerning microrole labels: for example the first argument of EAT is labelled as *eater*, whereas the same argument of DRINK is tagged as *drinking person.* These inconsistencies are partly related to the addition of new meanings discussed above: EAT is one of the core 80 verb meanings, whereas DRINK has been added to ValPaL at a later stage.

In a separate section of the database, microroles (but not all of them, see Section 2.2) are grouped into roles. The latter are also employed in the role frame provided for each verb meaning (see, in Figure 3, the role frame for FEAR). Such roles partly overlap with the set of argument types, partly add to them. A list of roles can be found in a footnote in the guidelines downloadable from the ValPaL webpage, which leaves space for possible additions, and possibly modifications: "*We often use letters that can be thought of as standing mnemonically for particular roles (A: agent, P: patient, S: single central argument of intransitive verb, T: theme (of ditransitive verb), R: recipient (of ditransitive verb), L: location (including goal), I: instrument, E: experiencer, M: stimulus, X, Y, Z: other). No claims are associated with the use of these letters, and they could be replaced by other arbitrary variable symbols*." (Database Questionnaire Manual, fn. 6, https://valpal.info/database).

2.2 Some issues related to the ValPaL data

Some inconsistencies related to the new meaning addition and microroles have been discussed in Sections 2 and 2.1 above. In this section, we add to this, by elaborating on issues regarding data collection, alternation storage and labelling, microrole grouping and derived coding frame collection.

The data collected for the database has been elicited by contributors in different ways. Often, contributors were also native speakers of the language for which they were responsible, and heavily relied on their intuition for data collection. In other cases, they relied on their own fieldwork, or on data from previous works by themselves or by other authors. Only occasionally the data was collected from corpora.

The number of alternations listed varies widely across languages, ranging from 42 for English to 5 for Besta. This makes comparison complicated, as it may indicate that contributors stored alternations based on different levels of granularity. In addition, there is no consensus across contributions on how the same alternation is labelled: for example, the same alternation occurring with the meaning FILL whereby an instrumental adjunct is promoted to subject (as in *Water filled the tub*) is labelled 'Instrumental subject' in English, 'Instrument to subject alternation' in German and Russian, and 'Oblique subject' in Italian, making cross-linguistic comparison complicated.

Each verb meaning is assigned a role frame, with semantic roles covering a number of more fine-grained microroles (see Figure 3 in Section 2.1 and cf. Haspelmath & Hartmann 42-43; Malchukov 2015: 74). For example, the role frame for the meaning BRING is "A brings T to R" (see Section 2.1 for the role labels), possible microroles are *bringer, brought thing, bringing recipient, bring causer, bringing instrument*. Microroles have been added by contributors without specifying under which role label they should be grouped. So while in the case of BRING one finds *bringer* A, *brought thing* T*, bringing recipient* R*,* the remaining two microroles, *bring causer* and *bringing instrument* are not further specified (see the data in https://valpal.info/microroles).

Moreover, a number of derived coding frames are missing. For example, according to the ValPaL data, the Italian verb *caricare* regularly features 10 alternations: the so-called *Object omission, Passive, Reflexive passive, Locative alternation, Anticausative (coded), Indirect/dative reflexive, Impersonal reflexive, Causative, Impersonal of Reflexives, Impersonal passive.* Among these, only eight alternations are paired with their derived coding frame; for example, "2 > passV'.subj[2] (su+3) (daParteDi+1)" is the derived coding frame of the *Passive* alternation, as we discussed in Section 2.1. In the cases of the *Indirect/dative reflexive* and of the *Impersonal of reflexive* alternations, this piece of information is missing, which makes it hard for database users to understand the coding details of certain alternations.

3. New features in PaVeDa

The aim of PaVeDa is twofold. In the first place, more languages have been and will be added, starting with, but not limited to ancient Indo-European languages that have a modern counterpart already stored in ValPaL. This enables diachronic comparison and offers evidence for changes in valency patterns and alternations (Section 3.1). In the second place, an intermediate level of annotation to the original ValPal has been added, called "alternation class", which categorizes language-specific alternations into four cross-linguistic types. Because comparison is an essential part of our research, we added a dedicated tool to compare basic frames and alternations across all languages and between individual languages (Section 3.2). PaVeDa also aims to add the missing role labels to all microroles and correcting some discrepancies discussed above (Section 3.3).

3.1 Adding a diachronic dimension

To date, ancient Indo-European languages added to PaVeDa are Old Latin, Ionic-Attic Ancient Greek, Gothic, Old English, Classical Armenian and Old High German. Apart from Gothic, that does not have any modern descendent, four other languages have their modern counterpart already stored in ValPaL: Italian, English, Eastern Armenian and German. Because Ionic-Attic Ancient Greek did not have its modern counterpart already available, we also added Modern Greek to the database. The information on basic valency patterns and alternations included for ancient languages relies on corpus data. Old Latin is based on the Plautus' corpus, whereas a corpus of Classical Greek prose comprising orators, historians and Plato has been scrutinized for Ionic-Attic Ancient Greek[1]. The reference corpus for Gothic is the fourth-century translation of the Bible, traditionally attributed to the Gothic bishop Wulfila (see Zanchi & Tarsi 2021: 31-34)[2]. The corpus for Old English consists of both prose (e.g. Ælfric's Catholic Homilies and Bede's History of the English Church; see Taylor et al. 2003) and poetry (e.g. the Beowulf and the Anglo-Saxon Elegies; see Pintzuk & Plug 2002), and includes texts differing in period, genres, and dialect. For Classical Armenian the New Testament has been scrutinized. Finally, data for Old High German is based on the REA corpus (Krause and Zeldes 2016), limited to Old High German texts[3].

The corpora used for such languages differ in terms of corpus-size and genre; these differences are due to the fact that, even though these languages all qualify as corpus languages, the available corpora that survived up to the present time are very different, which makes corpus harmonization virtually impossible. Concerning data extraction, PaVeDa contributors adopt different methodologies. In the case of languages with small and close corpora such as Gothic and Old Latin, all the occurrences of verb lemmas selected have been analyzed. In case of large-corpus languages such as Ionic-Attic Ancient Greek all the occurrences of verb lemmas whose frequency in the reference corpus is lower than 100 occurrences have been analyzed, whereas, for verb lemmas with frequency higher than 100, a stratified random sample of 100 occurrences has been extracted. These 100 occurrences are assumed to contain instantiations of all alternations featured by a certain verb. Notably, this assumption has always been double-checked against reference dictionaries and grammars. All added ancient languages are cross-referenced to Glottolog. For this reason, we tried to adhere to Glottolog language names as close as possible, as in the case of Old Latin and Ionic-Attic Ancient Greek (Glottolog does not feature a generic label Ancient Greek, while the label Latin only refers to Late, Vulgar and Medieval Latin, cf. https://glottolog.org/resource/languoid/id/lati1261).

3.2 Issues brought about by the addition of ancient languages

Elicitation of data for ancient languages brings about a number of theoretical issues that have a more general scope. The most challenging issue is of course the impossibility to rely upon native speakers' judgments to rate the basicness of competing verbs for any given core meaning, let alone alternations. Following the methodology laid out in Zanchi & Tarsi (2021), we used a combination of morphological and frequency criteria to overcome this issue as detailed below.

Verb lemmas that are morphologically underived or that exhibit the simplest morphological structure are regarded as more basic (e.g. in Old Latin the verb *eō* is preferred over the preverbed *ad-eō* 'approach' for the meaning GO). If a verb is underived but is scarcely attested in the reference corpus, a derived verb is selected instead, provided that its number of occurrences is significantly higher. For example, for the meaning LIKE the derived Gothic verb *ga-leikan* (attested 20 times in the Gothic corpus) has been selected instead of *leikan* (one occurrence) because of its higher frequency. Frequency also drives the choice between verbal lemmas with comparable degrees of morphological complexity (e.g. in Old Latin for the meaning FEAR the verb *metuō*, 154 occurrences in the reference corpus, is preferred over *timeō*, 35 occurrences). In cases in which neither of these criteria is applicable, we decided to take into account the historical developments of the candidate lemmas, and possibly select more than one verb. For example, for the meaning EAT both the Gothic verbs *matjan* and *itan* were included in the database, as the latter, despite being less-frequent than the former in the Gothic corpus, continues in several modern Germanic languages (e.g. English *eat*, German *essen*).

Frequency is also disfavored in cases in which the more frequent verb for a given meaning is polysemous. Take as an example the two competing Old Latin lemmas *petō* and *poscō* for the meaning ASK FOR. Despite its lower frequency, *poscō* has

[1] Corpora for Old Latin and Ionic-Attic Ancient Greek have been scrutinized with the Perseus Digital Library (https://www.perseus.tufts.edu/hopper/).

[2] The Gothic Gospels are available at the PROIEL project and Wulfila project websites (PROIEL Project: http://foni.uio.no:3000/sources/11; Wulfila project: http://www.wulfila.be.

[3] For the REA corpus see https://www.deutschdiachrondigital.de/rea/ and http://dsh.oxfordjournals.org/content/31/1/118 available at https://korpling.german.hu-berlin.de/annis/ddd. Notably, the REA corpus also contains texts in Old Saxon and Old Low Franconian, which have been left out from our account. This has been easily done, as texts can be selected individually in REA. Old Latin data was collected by Martina Giuliani (University of Pavia / University of Bergamo); Chiara Zanchi (University of Pavia) and Guglielmo Inglese (University of Turin) are responsible for Ionic-Attic Ancient Greek; Matteo Tarsi (Uppsala University) and Chiara Zanchi added the Gothic data. The Old English data was collected by Martina Giarda (University of Pavia / University of Bergamo), and the Old High German one by Giacomo Bucci (Ghent University). Petr Kocharov took care of the Classical Armenian section of the database. The addition of corpus data for Modern Russian was carried out by Erica Pinelli, Irina Parshina and Maria Bocharova. Lucrezia Carnesale collaborated in the creation of the database.

been selected instead of *petō,* because its semantics better fits the meaning ASK FOR. The verb *petō* is highly polysemous and is frequently used with the meanings ‘assault, attack’ and ‘go, travel toward’, along with expressing requests. As argued by Inglese (2021: 142) “verbs that are primarily associated with a given meaning are preferred over those that express that meaning only secondarily and/or metaphorically”. Selecting *poscō* would have forced us to analyze all the occurrences of the lemma to look for those instantiating the meaning relevant for the database.

Of course, especially with languages such as Gothic for which only a limited corpus is available, missing attestation of some verb meanings or constructions does not necessarily reflect a gap in a language’s lexicon or grammar but it may reflect a gap in the corpus. The same is true for Old Latin whose reference corpus is the collection of Plautus’ comedies (see Section 3.1). For verb meanings not sufficiently represented because of corpus selection, additional corpora (e.g. Terence’s corpus for Old Latin) and lexicographic resources have also been checked.

In spite of these challenges, using corpus data has an undoubted advantage over relying on the intuition of individual speakers, as corpora contain more than what is evident to speakers’ intuition, provide real usage-based occurrences and also data about their actual frequency. As Fillmore’s (1992: 35) puts it: “[…] every corpus I have had the chance to examine, however small, has taught me facts I couldn’t imagine finding out any other way”. We will return on this important point in Section 4.

In languages that do not rely on a large enough corpus of attestations it may be the case that some of the ValPaL verb meanings are not retrievable. In such cases, other verb meanings have been selected, to partly compensate for the gaps in coverage, that can reasonably be expected to elicit verbs with a comparable syntactic behavior to those which are not attested. For example, the Gothic section of the database does not comprise lemmas for the ValPaL core meanings BE A HUNTER, BLINK, BOIL, COUGH, FEEL COLD, HUG, PLAY and SMELL. In order to partially compensate these gaps, new meanings have been added, i.e. CRY, DIG, DRINK, FALL, GRIND and LIGHTEN. All new meanings are cross-referenced to Concepticon.

Corpus-based approaches also challenge the assumption that ValPaL core meanings are representative of the entire verbal lexicon, as some argument structure constructions are under-represented due to verb meanings selection. An example is the domain of experience in Old Latin. ValPaL core meanings fail to account for a group of Latin experiential verbs denoting negative emotions (e.g. *pudet* ‘be ashamed’), which show a peculiar argument structure construction (see Fedriani 2014 among others). These verbs are constructed impersonally: they are inflected in the third person singular active (rarely passive) form, without a fully-fledged syntactic subject in the nominative, and take two arguments: an accusative experiencer and a genitive stimulus. To also included Latin verbs featuring this construction in the database, five new verb meanings have been added to PaVeDa: BE ANNOYED, BE ASHAMED (cf. (1), (2)), DISPLEASE, HAVE PITY and REGRET[4].

(1) Verb meaning: BE ASHAMED
Old Latin verb: *pudet*
Microroles:
1. ashamed person
2. ashaming thing

Basic coding frame: 1-acc 2-gen V.3SG

(2) Example of the basic coding frame:

quoius	*me*	*nunc*
REL.GEN.SG	1SG.ACC	now
facti	*pudet*	
deed:GEN.SG	be_ashamed:PRS.3SG	

‘a deed which I am now ashamed of.’ (Plaut. *Bacch*. 1016)

As the addition of new meanings leads to the addition of new microroles and, ideally, should be extended to all languages in the database, such additions are discussed with the project coordinators and managed by them. All newly added meanings will also be externally cross-referenced with Concepticon.

The role of frequency in corpora cannot be underestimated, and has brought us to reconsider the way in which the data stored in ValPaL have been elicited and, more in general, how one should elicit data for modern languages and how the valency of a verb should be established. For this reason, we plan to add corpus data to languages already stored in ValPaL, following a usage-based notion of valency (see Section 4).

3.3 Alternation classes

In order to make cross-linguistic comparison easier, we added an intermediate level of alternations that we have called “alternation class”. Following Malchukov (2015: 96-103 and references therein) language-specific alternations have been classified into four coarse-grained groups: (i) Argument-decreasing; (ii) Argument-increasing; (iii) Argument-rearranging; and (iv) Argument identifying. Alternations affecting the number of verbs’ arguments have been marked either as Argument-decreasing or -increasing. As argument-decreasing strategy see the generic argument omission in Ionic-Attic Ancient Greek, as in (3) and (4).

(3) Verb meaning: EAT
Ionic-Attic Ancient Greek verb: *esthíō*
Basic coding frame:
1-nom V.act.subj[1] 2-acc

[4] All examples used in this paper are from the PaVeDa database. In case the new language employs a script different from the Latin one, the original text is provided, along with its transliteration, glosses and translation. When available, we plan to add links to external language resources indicating to the loci of the added examples.

Derived coding frame: 1-nom V.act.subj[1]

(4) Example of the generic argument omission alternation in Ionic-Attic Ancient Greek:
ὅτι *ἀηδῶς* *ἐσθίοι*
hóti *aēdôs* *esthíoi*
that unpleasantly eat.PRS.OPT.3SG
'That he eats unpleasantly.' (Xen. Mem. 3.13.2.1)

An argument-augmenting strategy is the cognate/kindred argument alternation in Old English, shown in (5) and (6).

(5) Verb meaning: LIVE
Old English verb: *lifian*
Basic coding frame: 1-nom V.subj[1] (in 2-dat)
Derived coding frame: 1-nom V.subj[1] 4-acc-cognate (in 2-dat)

(6) Example of the cognate/kindred argument alternation in Old English:
Lifd *se*
live.IND.PRET.3SG DET.NOM.SG.M
mon *his*
man(M).NOM.SG POSS.3SG.M
liif *in* *micelre*
life(N).ACC.SG in great.DAT.SG.F
forhæfdnisse
abstinence(F).DAT.SG
'The man lived a life of great abstinence.' (Bede_4:26.350.6.3521_ID)

Alternations implying a change in the encoding of verbs' arguments but not in their number are Argument-rearranging. An example is the partitive alternation attested in Ionic-Attic Ancient Greek in (7) and (8).

(7) Verb meaning: CUT
Ionic-Attic Ancient Greek verb: *témnō*
Basic coding frame: 1-nom V.act.subj[1] 2-acc (3-dat)
Derived coding frame: 1-nom V.act.subj[1] 2-gen

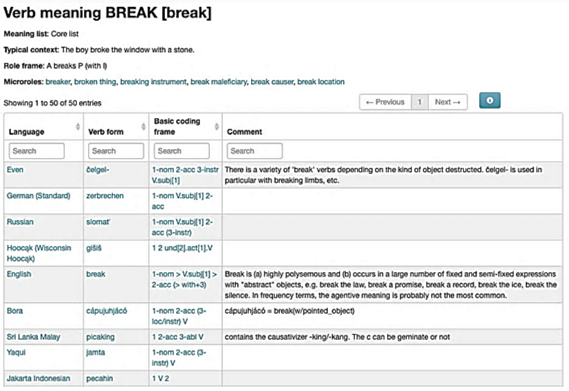
(8) Example of the partitive alternation in Ionic-Attic Ancient Greek:
τῆς *ὕλης*
tês *húlēs*
ART.GEN.F wood(F).GEN
τέμνοντα
témnonta
cut.AOR.PTCP.ACC
'Having cut wood' (Xen. Cyneg. 2.9.3)

Finally, the class Argument-identifying has been assigned to reflexive and reciprocal alternations, see e.g. the direct reflexive alternation in Old Latin shown in (9) and (10).

(9) Verb meaning: COVER
Latin verb: *tegō*
Basic coding frame: 1-nom 2-acc V.subj[1]
Derived coding frame: 1=2-nom 1=2-acc-refl V.subj[1=2]

(10) Example of the direct reflexive alternation in Old Latin:
capite *se*
top(N):ABL.SG REFL.ACC
totum *tegit*
entire(N):ACC.SG cover:IND.PRS.3SG
'He covers himself entirely with his top' (Plaut. Trin. 851)

Having added this level, which does not exist in the original ValPaL, we now have new options for comparison. In order to compare ancient languages with their modern counterpart, we have added it not only in the new languages stored in PaVeDa but also to some of the languages already stored in ValPaL and imported into PaVeDa, i.e. English, German, Italian and East Armenian. We can now look for all alternations belonging to one of the four groups in the relevant languages, or all alternations, again divided into the four groups under each verb meaning.
In addition, we implemented the option of directly comparing a verbal meaning, with basic frames and alternations in two given languages.
Let us take the verb meaning BREAK. Presently, ValPaL offers the option of visualizing the basic frames occurring in all languages.



Verb meaning BREAK [break]

Meaning list: Core list

Typical context: The boy broke the window with a stone.

Role frame: A breaks P (with I)

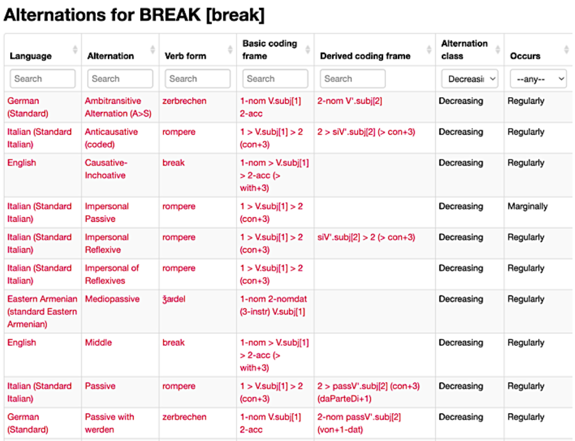
Microroles: breaker, broken thing, breaking instrument, break maleficiary, break causer, break location

Showing 1 to 50 of 50 entries

Language	Verb form	Basic coding frame	Comment
Search	Search	Search	Search
Even	čelgel-	1-nom 2-acc 3-instr V.subj[1]	There is a variety of 'break' verbs depending on the kind of object destructed. čelgel- is used in particular with breaking limbs, etc.
German (Standard)	zerbrechen	1-nom V.subj[1] 2-acc	
Russian	slomat'	1-nom V.subj[1] 2-acc (3-instr)	
Hoocąk (Wisconsin Hoocąk)	gišiš	1 2 und[2].act[1].V	
English	break	1-nom > V.subj[1] > 2-acc (> with+3)	Break is (a) highly polysemous and (b) occurs in a large number of fixed and semi-fixed expressions with "abstract" objects, e.g. break the law, break a promise, break a record, break the ice, break the silence. In frequency terms, the agentive meaning is probably not the most common.
Bora	cápujuhjácó	1-nom 2-acc (3-loc/instr) V	cápujuhjácó = break(w/pointed_object)
Sri Lanka Malay	picaking	1 2-acc 3-abl V	contains the causativizer -king/-kang. The c can be geminate or not
Yaqui	jamta	1-nom 2-acc (3-instr) V	
Jakarta Indonesian	pecahin	1 V 2	

Figure 4: Basic frames of BREAK

To this, we added the option of visualizing all attested alternations (for BREAK they are 250), or to select those belonging to one of the four groups at the intermediate level. In Figure 5 we show all argument -decreasing alternation contained in the database for the meaning BREAK.

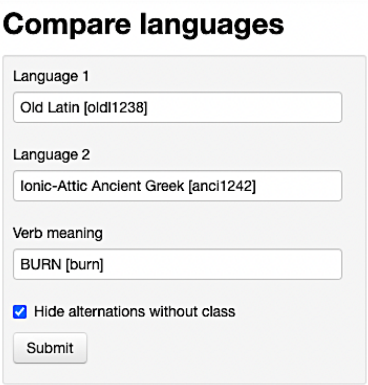


Alternations for BREAK [break]

Language	Alternation	Verb form	Basic coding frame	Derived coding frame	Alternation class	Occurs
Search	Search	Search	Search	Search	Decreasi	--any--
German (Standard)	Ambitransitive Alternation (A>S)	zerbrechen	1-nom V.subj[1] 2-acc	2-nom V'.subj[2]	Decreasing	Regularly
Italian (Standard Italian)	Anticausative (coded)	rompere	1 > V.subj[1] > 2 (con+3)	2 > siV'.subj[2] (> con+3)	Decreasing	Regularly
English	Causative-Inchoative	break	1-nom > V.subj[1] > 2-acc (> with+3)		Decreasing	Regularly
Italian (Standard Italian)	Impersonal Passive	rompere	1 > V.subj[1] > 2 (con+3)		Decreasing	Marginally
Italian (Standard Italian)	Impersonal Reflexive	rompere	1 > V.subj[1] > 2 (con+3)	siV'.subj[2] > 2 (> con+3)	Decreasing	Regularly
Italian (Standard Italian)	Impersonal of Reflexives	rompere	1 > V.subj[1] > 2 (con+3)		Decreasing	Regularly
Eastern Armenian (standard Eastern Armenian)	Mediopassive	ǯardel	1-nom 2-nomdat (3-instr) V.subj[1]		Decreasing	Regularly
English	Middle	break	1-nom > V.subj[1] > 2-acc (> with+3)		Decreasing	Regularly
Italian (Standard Italian)	Passive	rompere	1 > V.subj[1] > 2 (con+3)	2 > passV'.subj[2] (con+3) (daParteDi+1)	Decreasing	Regularly
German (Standard)	Passive with werden	zerbrechen	1-nom V.subj[1] 2-acc	2-nom passV'.subj[2] (von+1-dat)	Decreasing	Regularly

Figure 5: Argument-decreasing alternations for BREAK

Comparison between two languages allows visualizing the basic frame and all the alternations that occur in those two languages (see Figure 6).



Compare languages

Language 1

Old Latin [oldl1238]

Language 2

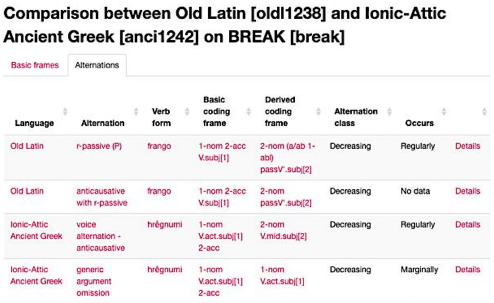
Ionic-Attic Ancient Greek [anci1242]

Verb meaning

BURN [burn]

☑ Hide alternations without class

Figure 6: BREAK in Old Latin and Ionic-Attic Ancient Greek



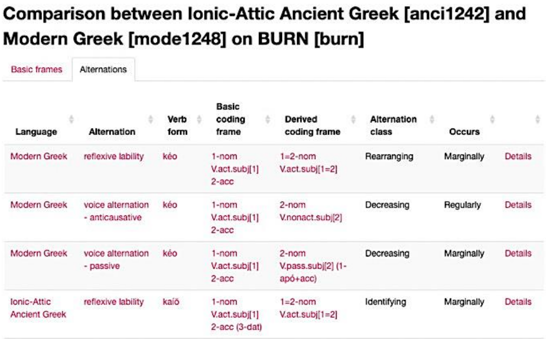
Comparison between Old Latin [oldl1238] and Ionic-Attic Ancient Greek [anci1242] on BREAK [break]

Language	Alternation	Verb form	Basic coding frame	Derived coding frame	Alternation class	Occurs	
Old Latin	r-passive (P)	frango	1-nom 2-acc V.subj[1]	2-nom (a/ab 1-abl) passV'.subj[2]	Decreasing	Regularly	Details
Old Latin	anticausative with r-passive	frango	1-nom 2-acc V.subj[1]	2-nom passV'.subj[2]	Decreasing	No data	Details
Ionic-Attic Ancient Greek	voice alternation - anticausative	hrḗgnumi	1-nom V.act.subj[1] 2-acc	2-nom V.mid.subj[2]	Decreasing	Regularly	Details
Ionic-Attic Ancient Greek	generic argument omission	hrḗgnumi	1-nom V.act.subj[1] 2-acc	1-nom V.act.subj[1]	Decreasing	Marginally	Details

Figure 7: BREAK in Old Latin and Ionic-Attic Ancient Greek

In Figure 7 we compare the alternations of Ionic-Attic Ancient Greek *hrḗgnumi* and Old Latin *frangō*. We can remark the mediopassive voice encodes the anticausative alternation in both languages, but it encodes the passive only in Latin.

Comparing an ancient language with its modern counterpart also leads to interesting remarks. In Figure 8 we compare the alternations of the Ionic-Attic Ancient Greek verb *kaíō* Modern Greek *kéo* ‘burn’. We can see that the main function of the mediopassive voice remains the encoding of the anticausative alternation, while encoding of the passive voice remains marginal at both languages stages (Luraghi and Mertyris 2021).

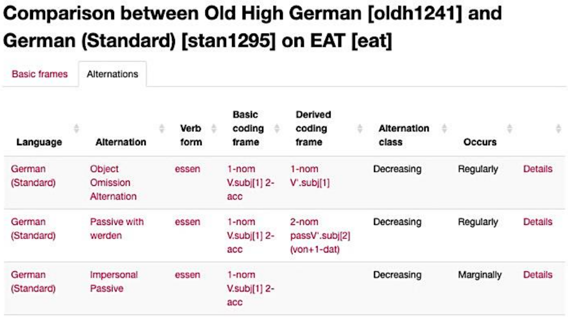


Comparison between Ionic-Attic Ancient Greek [anci1242] and Modern Greek [mode1248] on BURN [burn]

Language	Alternation	Verb form	Basic coding frame	Derived coding frame	Alternation class	Occurs	
Modern Greek	reflexive lability	kéo	1-nom V.act.subj[1] 2-acc	1=2-nom V.act.subj[1=2]	Rearranging	Marginally	Details
Modern Greek	voice alternation - anticausative	kéo	1-nom V.act.subj[1] 2-acc	2-nom V.nonact.subj[2]	Decreasing	Regularly	Details
Modern Greek	voice alternation - passive	kéo	1-nom V.act.subj[1] 2-acc	2-nom V.pass.subj[2] (1-apó+acc)	Decreasing	Marginally	Details
Ionic-Attic Ancient Greek	reflexive lability	kaíō	1-nom V.act.subj[1] 2-acc (3-dat)	1=2-nom V.act.subj[1=2]	Identifying	Marginally	Details

Figure 8: BURN in Ancient and Modern Greek

In Figure 9 we compare the verb meaning EAT in Old High German and Modern Standard German.

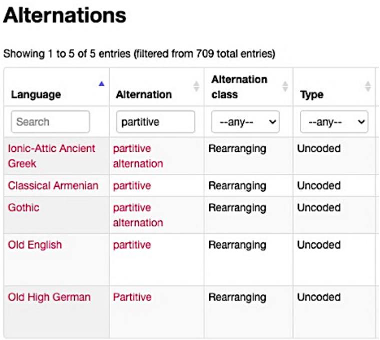


Comparison between Old High German [oldh1241] and German (Standard) [stan1295] on EAT [eat]

Language	Alternation	Verb form	Basic coding frame	Derived coding frame	Alternation class	Occurs	
German (Standard)	Object Omission Alternation	essen	1-nom V.subj[1] 2-acc	1-nom V'.subj[1]	Decreasing	Regularly	Details
German (Standard)	Passive with werden	essen	1-nom V.subj[1] 2-acc	2-nom passV'.subj[2] (von+1-dat)	Decreasing	Regularly	Details
German (Standard)	Impersonal Passive	essen	1-nom V.subj[1] 2-acc		Decreasing	Marginally	Details

Figure 9: EAT in Old High German and Modern Standard German

In Old High German the partitive alternation occurs, which has disappeared in Modern German. Indeed, this particular Argument-rearranging alternation, which involves the partitive genitive (or the ablative case in Classical Armenian) as direct object case is typical of ancient, as opposed to modern Indo-European languages, coherently with the data in Figure 10 (see Luraghi and Kittilä 2014).



Alternations

Showing 1 to 5 of 5 entries (filtered from 709 total entries)

Language	Alternation	Alternation class	Type
Search	partitive	--any--	--any--
Ionic-Attic Ancient Greek	partitive alternation	Rearranging	Uncoded
Classical Armenian	partitive	Rearranging	Uncoded
Gothic	partitive alternation	Rearranging	Uncoded
Old English	partitive	Rearranging	Uncoded
Old High German	Partitive	Rearranging	Uncoded

Figure 10: The partitive alternation

3.4 Semantic roles and microroles

As we said in Section 2.2 labels for semantic roles are introduced in a footnote of the guidelines, and it is explicitly stated that they are arbitrary. When one looks at the classification of microroles according to their correspondence to a role, one can see a number of discrepancies. Some of them are connected with the use of the label S, defined as *single central argument of intransitive verb*. Indeed, this definition is problematic because it refers to a syntactic, rather than semantic property. In particular, experiential verbs are often monovalent, so their subjects should be labelled S, but as the label E *experiencer* is also in the list, in the database they are variously labelled S (as in the case of FEEL COLD) or E (as in the case of BE HUNGRY and BE SAD, whose role frame also contains a single argument). Similarly, the single argument of motion verbs is usually variously assigned the role S (e.g. the verb meaning GO) or A (the verb meanings RUN and JUMP). Other discrepancies are shown by the use of the label R (recipient of a ditransitive verb). While the third participant of verb meanings such as GIVE and

BRING (*bringing recipient*, *giving recipient*, respectively) is assigned to R (cf. Section 2.2), the third participant of the meaning SEND is instead assigned to X: other.

These discrepancies have not allowed us to implement a further level of comparison among semantic roles yet. This comparative level will allow users to visualize how a certain semantic role is encoded in the project languages. To reach this goal, we are presently trying to unify role assignment to microroles.

4. PaVeDa in the (near) future

As for the diachronic dimension, we plan to add other ancient Indo-European languages to PaVeDa. Recruited project members have already started working on Old Italian, Old Church Slavonic, Old Icelandic, Sanskrit, Old Irish and Hittite.

Thus, besides including ancient Indo-European languages that already have a modern counterpart in ValPaL (e.g. Old Icelandic - Icelandic) we will also include languages for which no modern descendent is stored in ValPaL, as is the case of Sanskrit or Old Irish. In such cases, we plan to also add modern counterparts and have already recruited contributors for Hindi and Modern Irish. We aim to have all sub-branches of the Indo-European language family stored in the database in order to allow employing the data for syntactic reconstruction, and reconstruct valency patterns and alternations for the proto-language (for previous efforts in this direction, see e.g. Barðdal and Smitherman 2013, Barðdal and Eythorsson 2016).

In regard to data coverage, we plan to include languages from families that are currently not stored in ValPaL, in particular Uralic and Turkic: our contributors are currently working on Finnish, Hungarian, Turkish and Chuvash, as well as from language families that are currently underrepresented, such as Afro-Asiatic (only Modern Standard Arabic is included in ValPaL). Increasing the number of Afro-Asiatic languages will also enable us to expand diachronic research outside the Indo-European languages: our contributors are currently working on Modern and Biblical Hebrew, and we have plans to further include diachronically diverse Arabic varieties.

Moreover, we are currently working on revising the data of some modern languages in light of corpus-based evidence provided by reference corpora. It is important to stress that our decision to add corpus data both to the modern languages that we have started adding (such as Modern Greek) and to the languages already stored in ValPaL has been prompted by our work with ancient languages. As we remarked in Section 3.2, working with a closed corpus may have limitations, but it also provides real data from language usage rather than data specifically elicited by a linguist from his/her native speaker intuition. Hence, work with ancient languages has had an impact on our view on how modern spoken languages should be investigated with concrete consequences on our methodology.

Up to now, Russian data has been partly revised by one of our project members based on data from the *Russian National Corpus* (available at https://ruscorpora.ru/en/), and discrepancies have indeed emerged from ValPaL examples coming from native speaker intuition and what is actually contained in corpora. For example for the verb *slomat'* 'break' not all alternations listed in ValPaL have been found in the *Russian National Corpus*; conversely, for the verb meaning LOOK AT the verb *smotret'* is given with no alternations, but in the corpus our contributors found the reflexive passive, as in (11).

(11) *Fil'm* *smotritsja*
film.SG.NOM watch.PRES.3SG.REFL
očen' *legko.*
very easily
'The movie is very easy to watch.'

In addition, while for various verbs possible alternations are listed that involve different verbal prefixes, e.g. under the meaning LOAD *nagruzit'* is used for the basic coding frame, but for the 'Prefixal Goal-Instrumental alternation' the verb *zagruzit'* is used. Following the same approach, for *smotret'* one could also add the participial passive alternation, which is documented in the corpus again in connection with a different prefix, *osmotret'*, but the contributors of ValPaL failed to do so.

To enhance the comparative possibilities offered by our database, we will further group languages specific alternations in a more fine-grained layer of 'comparative concepts' (Haspelmath 2010) describing alternation types such as 'passive', 'antipassive', 'applicative', and so forth. For coded alternations, we will build on the taxonomy proposed in Haspelmath (2022), while for uncoded alternations we will try to identify and correct the inconsistencies of the type described in Section 2.2.

So far, we have described implemented and planned comparative visualization for verb meanings and for functional units, such as alternations and semantic roles. Our last goal for the near future of PaVeDa is to introduce a lemma-based comparative visualization option, which will allow tracking whether and how cognate verbs change their valency patterns and alternations over time. This is possible as our contributors for ancient Indo-European languages have been asked to indicate cognates of the basic verbs they choose to include in the database.

5. Conclusion

This paper documents the work that has been done so far to create a new resource, PaVeDa, which is specifically designed for cross-linguistic and diachronic comparison of verb valency classes and alternations. Building on the ValPaL database, we implemented modifications regarding language coverage, data elicitation and database structure.

As for language coverage, to date we have added six ancient and one modern Indo-European languages, for a total of nine new meanings, 46 new microroles, 211 new coding frames. Two new options for searching the database have been implemented, one that allows to visualize simultaneously all alternations stored in the database for each verb meaning across all languages, and a second one that allows direct comparison of the alternations found in two given languages for each verb meaning.

Further plans concern the addition of other, both ancient and modern languages, as well as corpus data for all languages, including those stored in ValPaL, in order to have real, usage-based data on valency patterns and alternations, and minimize the impact of constructed data, based on native speaker intuition of individual researchers.

Finally, we are planning to implement a comparison option based on etymological information (that has been annotated for ancient languages but not yet uploaded into the database) to make possible tracking changes in valency patterns and alternations over time.

Our research shows how working with ancient languages may also bring about a change of perspective on the methodology adopted for research on modern languages, as in the case of favoring corpora over native speaker intuition as source for data elicitation.

Concerning the relation between PaVeDa and ValPaL, while the main goal is language comparison for both databases, we view diachronic comparison as equally important as typological comparison. In this regard, PaVeDa should not simply be viewed as an enhanced version of ValPaL, but as a new and independent resource in its own right, and a completely new resource for what concerns ancient languages.

6. Bibliographical References

Aldai, G. and Wichmann, S. (2018). Statistical observations on hierarchies of transitivity. *Folia Linguistica*, 52(2):249–281.

Barðdal, J. and Smitherman, T. (2013). The quest for cognates: a reconstruction of oblique subject constructions in proto-Indo-European. *Language dynamics and change*, 3(1):28–67.

Biagetti, E., Brigada Villa, L., Zanchi, C., and Luraghi S. (2023a). Enhancing the semantic and conceptual description of Ancient Greek verbs in WordNet with VerbNet and FrameNet: a treebank-based study. In *Papers from the Annual International Conference "Dialogue"*, Vol. 22 (Supplementary volume), pages 1009–1020.

Biagetti, E., Zanchi, C., and Luraghi, S. (2023b). Linking the Sanskrit WordNet to the Vedic Dependency Treebank: a pilot study. In *Proceedings of the 12th Global Wordnet Conference*, pages 77–83, University of the Basque Country, Donostia – San Sebastian, Basque Country. Global Wordnet Association.

Eythórsson, Th. and Barðdal, J. (2016). Syntactic reconstruction in Indo-European: State of the art. *Veleia*, 33:83–102.

Fedriani, C. (2014). *Experiential predicates in Latin.* Brill, Leiden.

Haspelmath, M. (2010). Comparative concepts and descriptive categories in cross-linguistic studies. *Languages*, 86(3):663–687.

Haspelmath, M. (2022). Valency and voice constructions (https://lingbuzz.net/lingbuzz/005941).

Haspelmath, M. and Hartmann, I. (2015). Comparing verbal valency across languages. In A. Malchukov & B. Comrie (Eds.), *Valency classes in the world's languages*. Berlin & New York: Mouton de Gruyter, pp. 41–72.

Haug, D. T.T. (2012). Syntactic conditions on null arguments in the Indo-European Bible translations. *Acta Linguistica Hafniensia*, 44(2):129–141.

Inglese G. (2021). Anticausativization and basic valency orientation in Latin. In S. Luraghi & E. Roma (Eds.), *Valency over Time: Diachronic Perspectives on Valency Patterns and Valency Orientation*. Berlin & New York: Mouton de Gruyter, pp. 133–168.

Keydana, G. and Luraghi, S. (2012). Definite referential null objects in Vedic Sanskrit and Ancient Greek. *Acta Linguistica Hafniensia*, 44(2):116–128.

Levin, B. (1993). *English verb classes and alternations*. University of Chicago Press, Chicago.

Luraghi, S. (2003). Definite referential null objects in Ancient Greek. *Indogermanische Forschungen* 108:167–194.

Luraghi, S. and Kittilä, S. (2014). Typology and diachrony of partitive case markers. In S. Luraghi & T. Huumo (Eds.), *Partitive cases and related categories*. Berlin & New York: Mouton de Gruyter, pp. 17–62.

Luraghi, S. and Mertyris, D. (2021). Basic valency in diachrony: from Ancient to Modern Greek. In S. Luraghi & E. Roma (Eds.), *Valency over Time: Diachronic Perspectives on Valency Patterns and Valency Orientation*. Berlin & New York: De Gruyter, pp. 169–208.

Malchukov, A. (2015). Valency classes and alternations: parameters of variation. In A. Malchukov & B. Comrie (Eds.), *Valency classes in the world's languages*. Berlin & New York: Mouton de Gruyter, pp. 73–130.

Malchukov, A. and Comrie, B. (2015). *Valency classes in the world's languages*. Berlin & New York: Mouton de Gruyter.

Mambrini, F., Passarotti, M. C., Litta Modignani Picozzi, E. M. G., and Moretti, G. (2021). Interlinking Valency Frames and WordNet Synsets in the LiLa Knowledge Base of Linguistic Resources for Latin. In *Proceedings of the 17th International Conference on Semantic Systems*, pages 16–28, Amsterdam, The Netherlands.

McGillivray, B., Passarotti, M. C., and Ruffolo, P. (2009). The Index Thomisticus Treebank Project: Annotation, Parsing and Valency Lexicon. *TAL*, 50(2):103–127.

McGillivray, B. and Passarotti, M. C. (2015). Accessing and using a corpus-driven Latin Valency Lexicon. In G. V. M. Haverling (Ed.), *Latin Linguistics in the Early 21st Century. Acts of the 16th International Colloquium on Latin Linguistics*, pages 289–300, Uppsala, Sweden, 6–11 June.

McGillivray, B. and Vatri, A. (2015). Computational valency lexica for Latin and Greek in use: a case study of syntactic ambiguity. *Journal of Latin Linguistics*, 14(1):101–126.

Say, S. (2014). Bivalent Verb Classes in the Languages of Europe: A Quantitative Typological Study. *Language dynamics and change*, 4(1): 116–166.

Sausa, E. and Zanchi, C. (2015). Non-accusative null objects in the Homeric Dependency Treebank. In F. Mambrini, M. Passarotti, & C. Sporleder (Eds.), *Proceedings of the Workshop on Corpus-Based Research in the Humanities* (CRH), pages 107–116, Warsaw, Poland, 10 December.

Zanchi, C., Sausa, E. and Luraghi, S. (2018). HoDeL, a Dependency Lexicon for Homeric Greek: issues and perspectives. In P. Cotticelli & F. Giusfredi (Eds.), *Proceedings of Formal Representation and Digital Humanities*. Cambridge: Cambridge Scholars Publishing, pp. 230–256.

Zanchi, C., Luraghi, S. and Combei, C. R. (2022). PaVeDa – Pavia Verbs Databse: Challenges and perspectives. In *Proceedings of the 4th workshop on research in computational linguistic typology and multilingual NLP*, pages 99-102. Seattle, Washington, Association for Computational Linguistics.

Zanchi, C. and Tarsi, M. (2021). Valency patterns and alternations in gothic. In S. Luraghi & E. Roma (Eds.), *Valency over Time: Diachronic Perspectives on Valency Patterns and Valency Orientation*. Berlin & New York: Mouton de Gruyter, pp. 31–88.

7. Language Resource References

Crane, G. R. (Ed.). *Perseus Digital Library*. Tufts University. Accessed on 22 February 2024.

Dag, T. T. H. and Jøhndal, M. L. (2008). Creating a Parallel Treebank of the Old Indo-European Bible Translations. In C. Sporleder & K. Ribarov (Eds.), *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, pages 27–34. http://dev.syntacticus.org/proiel.html#downloads. Accessed on 22 February 2024.

Hartmann, I., Haspelmath, M., and Taylor, B. (2013). *Valency Patterns Leipzig*. Leipzig: Max Planck Institute for Evolutionary Anthropology.

Krause, Th. and Zeldes, A. (2016): ANNIS3: A new architecture for generic corpus query and visualization. In *Digital Scholarship in the Humanities* 31(1):118–139.

List, J. M., Tjuka, A., van Zantwijk, M., Blum, F., Ugarte, C. B., Rzymski, Ch., Greenhill, S., and Forkel, R. (Eds.) (2023). CLLD Concepticon 3.1.0 [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7777629. Accessed 24 February 2024.

Pintzuk, S. and Plug, L. (2002). *The York Helsinki Parsed Corpus of Old English Poetry*. Department of Linguistics, University of York. Oxford Text Archive, first edition; http://www.users.york.ac.uk/~lang18/pcorpus.html. Accessed on 22 February 2024.

Passarotti, M. C., Gonzalez Saavedra, B. and Onambele Manga, C. L. (2016). Latin Vallex. A Treebank-based Semantic Valency Lexicon for Latin. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016*), pages 2599–2606.

RNM = The Russian National Corpus. Плунгян, В. А. Резникова, Т. И., Сичинава Д. В. (2005). Национальный корпус русского языка: общая характеристика. Научно-техническая информация. Сер. 2. № 3. С. 9—13. [Plungjan, V. A., Rezhnikova, T. I., Sichinava D. V. (2005). *Natsionalnyj corpus russkogo jazyka: obshchaja kharakteristika. Nauchno-tekhnicheskaja informats*]

Say, S. (2020). BivalTyp: Typological database of bivalent verbs and their encoding frames. https://www.bivaltyp.info. Accessed on 22 February 2024.

Taylor, A., Warner, A., Pintzuk S., and Beths, F. (2003). *The York Toronto Helsinki Parsed Corpus of Old English Prose (YCOE)*. Department of Linguistics, University of York. Oxford Text Archive, first edition. http://wwwusers.york.ac.uk/~lang22/YcoeHome1.htm. Accessed on 22 February 2024.

Zanchi, C. (2021). The Homeric Dependency Lexicon. What it is and how to use it. Journal of Greek Linguistics 21(2):263–297. https://hodel.unipv.it/hodel-res. Accessed on 22 February 2024.

Zeige, L. E., Schnelle, G., Klotz, M., Donhauser, K. Gippert, J., and Lühr, R. (2022). Deutsch Diachron Digital. Referenzkorpus Altdeutsch. Humboldt-Universität zu Berlin. http://www.deutschdiachrondigital.de/rea/. DOI https://doi.org/10.34644/laudatio-dev-MiXVDnMB7CArCQ9CABmW. Accessed on 22 February 2024.

8. Acknowledgments

Research for this paper and for the creation of PaVeDa has been supported by European Union funding – NextGenerationEU – Missione 4 Istruzione e ricerca - componente 2, investimento 1.1" Fondo per il Programma Nazionale della Ricerca (PNR) e Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN)" progetto 20223XH5XM "Verbs' constructional patterns across languages: a multi-dimensional investigation" CUP F53D23004570006.

Finanziato dall'Unione europea NextGenerationEU Ministero dell'Università e della Ricerca Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

Development of robust NER Models and Named Entity Tagsets for Ancient Greek

Chiara Palladino*[*] , Tariq Yousef[†]
[*]Furman University, USA, chiara.palladino@furman.edu
[†]University of Southern Denmark, Denmark, yousef@sdu.dk

Abstract

This contribution presents a novel approach to the development and evaluation of transformer-based models for Named Entity Recognition and Classification in Ancient Greek texts. We trained two models with annotated datasets by consolidating potentially ambiguous entity types under a harmonized tagset. Then, we tested their performance with out-of-domain texts, reproducing a real-world use case. Both models performed very well under these conditions, with the multilingual model *Ancient Greek Alignment* being slightly superior. In the conclusion, we emphasize current limitations due to the scarcity of high-quality annotated corpora and to the lack of cohesive annotation strategies for ancient languages.

Keywords: Token Classifications, Transformer Models, Named Entities Recognition, Ancient Greek, NLP

1. Introduction

Named Entity Recognition (NER) is a key task in text analysis and information extraction, which includes extracting, classifying, and disambiguating Named Entities (NEs) occurring in texts. The resulting outputs, which typically consist of datasets of classified names or annotated texts, provide important contextual information to facilitate interpretation of a source, and to enhance further explorations of it. Despite the current innovations in the application of transformer models to this task in ancient languages, Ancient Greek NER is still relatively unexplored. In this contribution, we illustrate a workflow to train a robust transformer-based NER in Ancient Greek with existing annotated texts. We ensured a state-of-the-art performance by mapping different entity types onto universal types, and performed a new type of evaluation with out-of-domain texts, reproducing a real-world scenario that provides a reliable assessment of the model's performance. In the conclusion, we present quantitative and qualitative results, and emphasize that current limitations are not due to scarce performance in available models, but to the lack of cohesive strategies for annotating and classifying entities in ancient languages.

2. Related Work

The introduction of Neural Networks and Deep Learning models has been a radical innovation in the computational processing of texts. Deep Learning was revolutionized by the introduction of transformers (Vaswani et al., 2017), which can capture contextual information to improve understanding of the data and retrieve that information from large contexts, and have become the state-of-the-art for extraction and classification tasks. In ancient languages, workflows based on popular transformer-based models such as BERT (Devlin et al., 2019) and RoBERTa (Conneau et al., 2020) have been applied to a wide variety of tasks, such as POS tagging, authorship attribution, text alignment, automatic translation, and paleographic analysis (Sommerschield et al., 2023; Yousef et al., 2023c).

The task of NER, however, has remained relatively unexplored. In the case of Latin, LatinCy (Burns, 2023) and LatinBERT (Bamman and Burns, 2020) have been shown to outperform state-of-the-art Machine Learning methods of the previous generation when trained on the NER task (Beersmans et al., 2023). BERT-based models have also been applied to Medieval Latin corpora (Torres Aguilar, 2022) and Sumerian (Wang et al., 2022). Compared to Latin, NER in Ancient Greek is less well-resourced: Singh et al. 2021 developed a BERT-based monolingual language model trained on Ancient and Byzantine Greek that showed optimal performance on POS tagging for in-domain data, while Brennan Nicholson trained a BERT model to predict missing characters (Nicholson, 2020). Neither model, however, was trained on NER.

To overcome the lack of training data and language models for ancient languages, Yousef et al. 2023a developed an annotation projection pipeline based on the word level alignment to project NER annotation from the English translations to the original ancient Greek texts. Yousef et al. 2023b trained the first transformer-based model for NER in Ancient Greek, *Ancient Greek Alignment*. The model leveraged on an XLM-R-based multilingual model fine-tuned on the word-alignment task for Ancient Greek and other languages (Yousef et al., 2022a,b;

Yousef, 2023), and it was trained for NER using ad hoc annotated corpora, achieving an F1-score higher than 90% in training and through evaluation with in-domain texts. However, it showed a much lower performance with less represented categories, particularly place-names, and in the detection of multi-token entities. Furthermore, confusion in entity labeling and the use of different tagsets in the training data led to frequent errors of miscategorization in the output. In this contribution, we illustrate how we improved the training with additional annotated corpora and a generalized entity tagset. Moreover, we present a new strategy for model evaluation using an out-of-domain corpus: this provides a much clearer understanding of the actual performance of a model, and it more closely reproduces a real-world scenario.

3. Training Datasets and Tagset Harmonization

We trained the models on available annotated corpora in Ancient Greek. Out of 17 historical corpora surveyed by Ehrmann et al. 2024, only two are in ancient languages (Latin and Coptic, none in Ancient Greek). New Latin corpora have become available through the Corpus Burgundiae (Torres Aguilar et al., 2016) and the LASLA project (Beersmans et al., 2023), and in Sumerian (Bansal et al., 2021). There are only two sizeable annotated datasets in Ancient Greek, which are currently under release: the first one by Berti 2023, consists of a fully annotated text of Athenaeus' *Deipnosophists*, developed in the context of the Digital Athenaeus project [1]. The second one by Foka et al. 2020, is a fully annotated text of Pausanias' *Periegesis Hellados*, developed in the context of the Digital Periegesis project [2]. In addition, we used smaller corpora annotated by students and scholars on Recogito[3]: the *Odyssey* annotated by Kemp 2021; a mixed corpus including excerpts from the *Library* attributed to Apollodorus and from Strabo's *Geography*, annotated by Chiara Palladino; Book 1 of Xenophon's *Anabasis*, created by Thomas Visser; and Demosthenes' *Against Neaira*, created by Rachel Milio. Table 1 provides an overview of the datasets used in the training.

The main issue with annotated corpora is the lack of a cohesive tagset for the classification of named entities. There are no generalized guidelines to annotate Named Entities in ancient texts (Beersmans et al., 2023). Therefore, projects focusing on ancient names use custom tagsets and guidelines that are very specific to the corpus being annotated.

	Person	Location	NORP
Training Dataset			
Odyssey	2.469	698	0
Deipnosophists	14.921	2.699	5.110
Pausanias	10.205	8.670	4.972
Other Datasets	3.283	2.040	1.089
Total	**30.878**	**14.107**	**11.171**
Validation Dataset			
Xenophon	1.190	796	857

Table 1: An overview of the training and validation datasets. For convenience, we have grouped the smallest datasets together.

This problem is particularly crucial because the size of annotated corpora currently available is very small, and ambiguous entities tend to be treated in very different ways: models cannot be trained to optimal results if similar entities are tagged in completely different ways, especially if they belong to underrepresented categories. One of the biggest issues is the often arbitrary use of names of socio-ethnic groups, which are subject to metonymic readings (Poibeau, 2006) or used as proxies for physical locations: these cases are sometimes labeled as places, sometimes as "proxies", sometimes as groups or ethnics. Furthermore, there is no agreement on the classification of groups ("the Athenians") and indications of ethnicity ("Athenian"). Because these cases are strongly dependent on context and interpretation, they are one of the biggest sources of disagreement among annotators (Álvarez Mellado et al., 2021).

In this contribution, we are not proposing a new tagset for the annotation of Named Entities in Ancient Greek. Rather, we suggest a strategy to harmonize already available corpora through tag mapping. We mapped the tagsets used in each corpus onto a general set of entity types, following the model outlined by Burns 2023 for LatinCy, which is based on a simplified version of the OntoNotes v.5.0 release (Weischedel et al., 2013) [4]. The tagset includes the same tags used in LatinCy: PERson (people, including fictional), LOCation (which combined countries, cities and states with non-GPE locations, such as water bodies), and NORP (nationalities, religious, or political groups).

There are several reasons behind the choice of this general tagset. First of all, it ensures consistency with another model for an ancient language that has already been tested successfully for NER. Moreover, it allows more consistency by harmonizing project-specific labels, particularly in complicated cases such as ethnonyms and groups of people. Even though the OntoNotes release is

[1] https://www.digitalathenaeus.org/

[2] https://periegesis.org/

[3] https://recogito.pelagios.org/

[4] http://www.bbn.com/NLP/OntoNotes

based on English, the NORP tag is general enough to include both located groups of people and ethnonyms, but also political and religious organizations in the ancient world. Therefore, it can also be mapped onto a more traditional GRP tag, as proposed by Beersmans et al. 2023, who expanded upon the guidelines outlined by the Herodotos project (Erdmann et al., 2019). Romanello and Najem-Meyer 2022 do not consider located groups, but use the ORG tag for religious and military groups or modern organizations: while we did not encounter enough of these categories to address them specifically, they can be mapped onto our definition of NORP. Because of their intrinsic ambiguity, we decided to avoid context-dependent labeling for proxies (people-for-place: "the Spartans moved war to the Athenians") and methonymic readings (place-for-people: "Athens voted to expel Themistocles"): the former is treated as NORP being a located group, and the latter is tagged as it appears (LOC), without making inferences on its function. Table 4 provides the full list of concordances.

4. Models

We conducted various experiments using different combinations of training datasets and underlying transformer models. We utilized the *Ancient Greek BERT* model developed by Singh et al. 2021, (from now on, the "monolingual" model, or *Model_A*) [5] , and the *Ancient Greek Alignment* model (from now on, the "multilingual" model, or *Model_B*), an XLM-R-based multilingual model[6] fine-tuned on the word alignment task for ancient languages (Yousef et al., 2022a,b; Yousef, 2023). In Ex1 and Ex2, we utilized the Deipnosophists dataset with the monolingual and multilingual models, respectively. In Ex3, we utilized the Pausanias dataset with the monolingual model. In Ex4, we combined both datasets and used the monolingual model. In Ex5 and Ex6, we utilized all available datasets mentioned in Table 1 with the monolingual and multilingual models, respectively. In all experiments, we trained the models for 10 epochs, using 80% of the dataset for training and the remaining 20% for testing. Table 5 provides an overview of the training results.

After training, the models were evaluated with an out-of-domain corpus consisting of the first three books of Xenophon's *Hellenica*, annotated on Recogito by a domain expert. The tagset used in the annotation of Xenophon followed the same internal guidelines adopted by Chiara Palladino, Thomas Visser and Rachel Milio in the training phase. The tagset was subsequently mapped onto the general one, following the same strategy already applied to the rest of the training data. The complete dataset includes a total of 2843 annotated entities, with a larger number of PER entities and a similar quantity of LOC and NORP entitites, as shown in Table1. Table 6 reports the complete overview on the models performance on the validation datasets.

5. Results

Table 2 summarizes the performance of the two models on the test and validation datasets. In the validation stage, both models performed considerably well, showing that a robust training workflow with a tagset harmonization strategy leads to state-of-the-art results with out-of-domain texts, and confirming the reliability of both models on the NER task in a real-world scenario. In particular, the performance achieved with PER and NORP entities was very high in both cases, while for LOC entities it was generally lower. The multilingual model [7] performed better in almost all categories, with an overall F1 score of 93.32% and accuracy of 98.87% in validation and and F1 score of 89.41% and accuracy of 97.5% in training. Place names (LOC) are still the most challenging entity type, with the monolingual model[8] performing at 87.1% and the multilingual model at 88.8%.

The worse performance on LOC can be partly explained by their representation in the training data, as they correspond to about half of the personal names in our datasets. However, this does not explain the much better performance on NORP entities, which are even less represented in the training data, yet led to a high performance in the output. On the one hand, this shows the robustness of the NORP tag chosen for the evaluation, especially considering that ethnonyms and groups are one of the most challenging entity classes for automatic extraction. On the other hand, it suggests that place names need a more careful treatment at the stage of annotation and guidelines design. Both models are now available on HuggingFace

5.1. Qualitative Evaluation

For the qualitative evaluation, we utilized the multilingual model (*Model_B*.) Overall, the multilingual model correctly classified 1118 PER entities, 698 LOC entities, and 809 NORP entities, for a total of 2625 entities (Table 3). It missed 78 entities, and it miscategorized 134 entities in total, with LOC being by far the most frequent. The most frequent errors

[5] https://huggingface.co/pranaydeeps/Ancient-Greek-BERT

[6] https://huggingface.co/UGARIT/grc-alignment

[7] https://huggingface.co/UGARIT/grc-ner-xlmr

[8] https://huggingface.co/UGARIT/grc-ner-bert

		Test		Validation	
		Model_A (Ex 5)	Model_B (Ex 6)	Model_A (Ex 5)	Model_B (Ex 6)
LOC	precision	82.92%	83.33%	87.10%	**88.66%**
	recall	81.30%	81.27%	87.10%	**88.94%**
	f1	82.11%	82.29%	87.10%	88.80%
NORP	precision	87.10%	88.71%	92.82%	**94.76%**
	recall	90.81%	90.76%	93.42%	**94.50%**
	f1	88.92%	89.73%	93.12%	**94.63%**
PER	precision	92.61%	91.72%	**95.52%**	94.22%
	recall	92.94%	94.42%	95.21%	**96.06%**
	f1	92.77%	93.05%	**95.37%**	95.13%
Overall	precision	88.92%	88.83%	92.63%	**92.91%**
	recall	88.82%	89.99%	92.79%	**93.72%**
	f1	88.87%	89.41%	92.71%	**93.32%**
	accuracy	97.28%	97.50%	98.42%	**98.87%**

Table 2: Test and validation results of the top two models. Model_A represents the output of Experiment 5, a fine-tuned model based on the ancient Greek monolingual model (Singh et al., 2021), while Model_B represents the output of Experiment 6, a fine-tuned model based on the Ugarit multilingual model (Yousef et al., 2022a).

of classification concerned confusion between the LOC and NORP tags, as it is to be expected. Very rarely confusion occurred between PER and other tags, often being justified by ambiguity in the very lemma of the word or by the presence of foreign names, such as "Mania", which was misclassified as LOC. Entities that were not extracted included some recurring names, such as "Phyle" (8 times) and "Otys" (6 times). The ethnonym "Hellenikon" was not extracted 5 times. There was also a minority of cases where the model correctly identified entities that had been mistakenly omitted by the annotator, which leads us to believe that the results are even better than what the numbers suggest.

Overall, LOC names were most frequently involved in errors of extraction and miscategorization. Interestingly, however, some common nouns were extracted and correctly classified, that could be considered places, such as "doors", "islands", "isthmus", "river", and "acropolis". This presumably reflects the ways in which entity boundaries are established in the training data, where strings like "Phasis river" or "Ionic gulf" are often considered full names, even if the second word is lowercase. However, it is also true that common nouns like "isthmus" are often used in Greek sources to refer to specific places, such as the Isthmus of Corinth: therefore, it is difficult to establish what exactly constitutes an identifiable "place" in these cases. A similar phenomenon occurred with titles, such as "hipparchos" (which can also be a personal name) or "ephoros", and with socio-political organizations, such as "boule" or "demos". It should be noted, however, that these strings were not consistently extracted: this is presumably due to the internal inconsistency of the training data, where analogous instances may or may not have been annotated, whether as names, as part of multi-token entities, or as mentions of specific referents, depending on the project guidelines.

A related issue is represented by multi-token entities, such as "Olympian Zeus" or "Temple of Artemis": these are often not represented in sufficient number to be significant for training and evaluation, and are extremely difficult to annotate, because their boundaries are not always clear. They are also challenging to measure in quantitative evaluation. In our dataset, there were 15 recognizable multi-token entities, of which the model extracted and classified 9 in a coherent way, while 5 were not recognized, and one was dubious. In most cases, even if the entity extracted did not perfectly overlap with the gold standard, it made sense: for example, "Lyceum gymnasium" was counted as an error because the annotator only tagged "Lyceum", but it is a perfectly acceptable alternative name. A remarkable case regarded the "Makra Teiche" (the Long Walls of Athens), which appears lowercase in our text, but was extracted and classified by the model. In other words, there are cases that need to be considered individually and qualitatively in order to be properly assessed, as they often require strict guidelines to establish entity boundaries.

6. Conclusion and Limitations

In this paper, we have shown a workflow to train a robust NER model, whose performance is evaluated on out-of-domain texts, reproducing a realistic scenario of use for a tool of this kind. Our training strategy and tagset harmonization lead to state-of-the-art performance with the two available

		Model Output			
		O	PER	LOC	NORP
Gold St.	O	26,226	33	37	14
	PER	37	1,118	22	8
	LOC	33	29	698	35
	NORP	8	2	38	809

Table 3: Qualitative Evaluation Confusion (Error) Matrix. "O" represents non-entity tokens.

transformer-based models, with a slightly better performance shown in the multilingual model trained on the alignment task.

Despite the encouraging results, the potential of transformers for NER in Ancient Greek is still not fully exploited. It has been shown that even the most refined models, without ad hoc training and fine-tuning, perform poorly on several tasks on ancient and historical corpora (Sprugnoli et al., 2023; González-Gallardo et al., 2023). Transformers are very data-hungry and require a significant amount of annotations for optimal results. This is especially relevant for ancient languages, which are closed systems and, for the most part, significantly smaller corpora than modern languages. This fundamentally limits strategies for upsampling, training and fine-tuning.

Apart from the scattered nature of currently available tagsets, some issues remain unresolved. For example, place names are still underrepresented in annotated corpora. However, data availability is insufficient to explain bad model performance, as we have shown above. In general, place names seem to be especially challenging for annotation practices, more than personal and group names. For example, the definition of identifiable "places" sometimes goes beyond capitalized words; furthermore, it may be relevant for a project to tag common nouns that refer to locatable areas. Another issue is represented by multi-token entities, such as "Pythian Apollo" or "Erythraean Sea". In our dataset, we had too few of them to be significant to the evaluation. However, the problem resides once again in annotation practices, as it is often difficult to establish the boundaries of what constitutes a named entity.

In conclusion, we want to emphasize that current challenges in model training and evaluation are not to be attributed to the lack of highly performing models, but to the lack of best practices and documentation in the development of high-quality annotated datasets (Beersmans et al., 2023). This key issue affects the further development of annotation strategies and reliable tagsets: in fact, our mapping strategy was effective in containing potentially ambiguous cases, but it also limited the granularity of entity classification. For example, author names, nicknames and personal names are all grouped under one PER tag, but their different functions could be significant in the context of individual projects. Furthermore, other entity classes were not considered, such as events, objects, and languages. The future necessary steps include the implementation of an extended tagset according to a hierarchical structure, as outlined by Romanello and Najem-Meyer 2022: the hierarchical structure will ensure that existing tagsets can still be harmonized at least at the higher level, but it will also provide a foundation for more accurate annotated corpora in the future.

Acknowledgments

We are deeply grateful to the people who contributed annotations and datasets to make this project possible. In no particular order: Monica Berti (Digital Athenaeus Project, University of Leipzig), Josh Kemp (Odyssey Project, Furman University), Thomas Visser and Rachel Milio (University of Exeter), and the whole team of the Digital Periegesis project: Elton Barker, Anna Foka, Brady Kiesling, Kyriaki Konstantinidou, Linda Talatas.

7. Bibliographical References

David Bamman and Patrick J. Burns. 2020. Latin BERT: A Contextual Language Model for Classical Philology. ArXiv:2009.10053 [cs].

Rachit Bansal, Himanshu Choudhary, Ravneet Punia, Niko Schenk, Émilie Pagé-Perron, and Jacob Dahl. 2021. How Low is Too Low? A Computational Perspective on Extremely Low-Resource Languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 44–59, Online. Association for Computational Linguistics.

Marijke Beersmans, Evelien de Graaf, Tim Van de Cruys, and Margherita Fantoli. 2023. Training and Evaluation of Named Entity Recognition Models for Classical Latin. In *Proceedings of the Ancient Language Processing Workshop*, pages 1–12, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Monica Berti. 2023. Named Entity Recognition for a Text-Based Catalog of Ancient Greek Authors and Works. *Zenodo (CERN European Organization for Nuclear Research)*.

Patrick J. Burns. 2023. LatinCy: Synthetic Trained Pipelines for Latin NLP. ArXiv:2305.04365 [cs].

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mª Luisa Díez Platas, Salvador Ros Muñoz, Elena González-Blanco, Pablo Ruiz Fabo, and Elena Álvarez Mellado. 2021. Medieval Spanish (12th–15th centuries) named entity recognition and attribute annotation system based on contextual information. *Journal of the Association for Information Science and Technology*, 72(2):224–238.

Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. 2024. Named Entity Recognition and Classification in Historical Documents: A Survey. *ACM Computing Surveys*, 56(2):1–47.

Alexander Erdmann, Christopher Brown, Brian Joseph, Mark Janse, Petra Ajaka, Micha Elsner, and Marie-Catherine de Marneffe. 2016. Challenges and Solutions for Latin Named Entity Recognition. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 85–93, Osaka, Japan. The COLING 2016 Organizing Committee.

Alexander Erdmann, David Joseph Wrisley, Benjamin Allen, Christopher Brown, Sophie Cohen-Bodénès, Micha Elsner, Yukun Feng, Brian Joseph, Bèatrice Joyeux-Prunel, and Marie-Catherine de Marneffe. 2019. Herodotos-Project/Herodotos-Project-Latin-NER-Tagger-Annotation. Original-date: 2017-10-22T06:51:43Z.

Anna Foka, Elton Barker, Kyriaki Konstantinidou, Nasrin Mostofian, O. Cenk Demiroglu, Brady Kiesling, and Linda Talatas. 2020. Semantically geo-annotating an ancient greek "travel guide" itineraries, chronotopes, networks, and linked data. In *Proceedings of the 4th ACM SIGSPATIAL Workshop on Geospatial Humanities*, GeoHumanities '20, page 1–9, New York, NY, USA. Association for Computing Machinery.

Carlos-Emiliano González-Gallardo, Emanuela Boros, Nancy Girdhar, Ahmed Hamdi, Jose G. Moreno, and Antoine Doucet. 2023. Yes but.. Can ChatGPT Identify Entities in Historical Documents? ArXiv:2303.17322 [cs].

Joshua Kemp. 2021. Beyond Translation: Building Better Greek Scholars. *Pelagios Blog*.

Brennan Nicholson. 2020. Ancient-greek-char-bert.

Chiara Palladino, Maryam Foradi, and Tariq Yousef. 2021. Translation Alignment for Historical Language Learning: a Case Study. *Digital Humanities Quarterly*, 015(3).

Thierry Poibeau. 2006. Dealing with Metonymic Readings of Named Entities. In *The 28th Annual Conference of the Cognitive Science Society (CogSci 2006)*, pages 1962–1968, Vancouver, Canada. Cognitive Science Society.

Matteo Romanello and Sven Najem-Meyer. 2022. Guidelines for the Annotation of Named Entities in the Domain of Classics. Publisher: Zenodo.

Aleksi Sahala. 2021. *Contributions to Computational Assyriology*. Doctoral Thesis, Faculty of Arts, University of Helsinki, Helsinki.

Stefan Schweter, Luisa März, Katharina Schmid, and Erion Çano. 2022. hmbert: Historical multilingual language models for named entity recognition. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum (CEUR Workshop Proceedings)*.

Pranaydeep Singh, Gorik Rutten, and Els Lefever. 2021. A pilot study for BERT language modelling and morphological analysis for ancient and medieval Greek. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 128–137, Punta Cana, Dominican Republic (online). Association for Computational Linguistics.

Thea Sommerschield, Yannis Assael, John Pavlopoulos, Vanessa Stefanak, Andrew Senior, Chris Dyer, John Bodel, Jonathan Prag, Ion Androutsopoulos, and Nando de Freitas. 2023. Machine learning for ancient languages: A survey. *Computational Linguistics*, pages 703–747.

Rachele Sprugnoli, Francesco Mambrini, Marco Passarotti, and Giovanni Moretti. 2023. The Sentiment of Latin Poetry. Annotation and Automatic

Analysis of the Odes of Horace. *IJCoL. Italian Journal of Computational Linguistics*, 9(1). Number: 1 Publisher: Accademia University Press.

Sergio Torres Aguilar. 2022. Multilingual Named Entity Recognition for Medieval Charters Using Stacked Embeddings and Bert-based Models. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 119–128, Marseille, France. European Language Resources Association.

Sergio Torres Aguilar, Xavier Tannier, and Pierre Chastang. 2016. Named entity recognition applied on a data base of Medieval Latin charters. The case of chartae burgundiae. In *3rd International Workshop on Computational History (HistoInformatics 2016)*, Krakow, Poland.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Guanghai Wang, Yudong Liu, and James Hearne. 2022. Few-shot Learning for Sumerian Named Entity Recognition. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 136–145, Hybrid. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Ramshaw Lance, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0 LDC2013T19.

Haneul Yoo, Jiho Jin, Juhee Son, JinYeong Bak, Kyunghyun Cho, and Alice Oh. 2022. HUE: Pretrained Model and Dataset for Understanding Hanja Documents of Ancient Korea. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1832–1844, Seattle, United States. Association for Computational Linguistics.

Tariq Yousef. 2023. *Translation Alignment Applied to Historical Languages: Methods, Evaluation, Applications, and Visualization*. Ph.D. thesis, Leipzig University.

Tariq Yousef, Chiara Palladino, Gerhard Heyer, and Stefan Jänicke. 2023a. Named Entity Annotation Projection Applied to Classical Languages. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 175–182, Dubrovnik, Croatia. Association for Computational Linguistics.

Tariq Yousef, Chiara Palladino, and Stefan Jänicke. 2023b. Transformer-based Named Entity Recognition for Ancient Greek. In *Digital Humanities 2023. Book of Abstracts*, pages 420–422, Graz. Centre for Information Modelling - Austrian Centre for Digital Humanities.

Tariq Yousef, Chiara Palladino, and Farnoosh Shamsian. 2023c. Classical Philology in the Time of AI: Exploring the Potential of Parallel Corpora in Ancient Language. In *Proceedings of the Ancient Language Processing Workshop*, pages 179–192, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Tariq Yousef, Chiara Palladino, Farnoosh Shamsian, Anise d'Orange Ferreira, and Michel Ferreira dos Reis. 2022a. An automatic model and gold standard for translation alignment of ancient greek. In *Proceedings of the Language Resources and Evaluation Conference*, pages 5894–5905, Marseille, France. European Language Resources Association.

Tariq Yousef, Chiara Palladino, David J. Wright, and Monica Berti. 2022b. Automatic Translation Alignment for Ancient Greek and Latin. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 101–107, Marseille, France. European Language Resources Association.

Amir Zeldes and Lance Martin. 2020. Coptic Scriptorium - Entity Annotation Guidelines. Version 1.1.0.

Elena Álvarez Mellado, María Luisa Díez-Platas, Pablo Ruiz-Fabo, Helena Bermúdez, Salvador Ros, and Elena González-Blanco. 2021. TEI-friendly annotation scheme for medieval named entities: a case on a Spanish medieval corpus. *Language Resources and Evaluation*, 55(2):525–549.

8. Appendix

Corpus	Original Tag	OntoNotes Tag
Deipnosophists	Person	PER
	Place	LOC
	Ethnic	NORP
	Group	NORP
	Noclass	MISC
	title	MISC
	festival	MISC
	month	MISC
	language	MISC
	constellation	MISC
Pausanias	Place.proxy	NORP
	Place.regional	LOC
	Place.physical	LOC
	Place.mythical	LOC
	Place.material	LOC
	Person	PER
Other	Place	LOC
	Place.group	NORP
	Ethnonym	NORP
	Person	PER
	Person.group	NORP
	Author	PER
	Patronymic	PER

Table 4: Concordance table used to harmonize the main tagsets used in the training data.

		Ex 1	Ex 2	Ex 3	Ex 4	Ex 5	Ex 6
LOC	precision	87.10%	86.11%	77.78%	80.54%	82.92%	83.33%
	recall	87.31%	88.59%	78.78%	80.23%	81.30%	81.27%
	f1	87.21%	87.33%	78.28%	80.39%	82.11%	82.29%
NORP	precision	93.35%	93.68%	89.51%	90.76%	87.10%	88.71%
	recall	92.32%	95.55%	92.08%	92.48%	90.81%	90.76%
	f1	92.83%	94.60%	90.78%	91.61%	88.92%	89.73%
PER	precision	94.10%	95.18%	88.78%	92.05%	92.61%	91.72%
	recall	95.67%	97.20%	88.62%	92.34%	92.94%	94.42%
	f1	94.88%	96.18%	88.70%	92.19%	92.77%	93.05%
overall	precision	91.55%	92.86%	85.36%	88.45%	88.92%	88.83%
	recall	91.74%	94.73%	86.17%	88.90%	88.82%	89.99%
	f1	91.64%	93.79%	85.76%	88.67%	88.87%	89.41%
	accuracy	98.21%	98.93%	95.55%	97.22%	97.28%	97.50%

Table 5: Training results of all experiments.

		Ex 1	**Ex 2**	**Ex 3**	**Ex 4**	**Ex 5**	**Ex 6**
LOC	precision	86.15%	**89.69%**	81.91%	86.76%	87.10%	88.66%
	recall	75.35%	75.89%	**91.02%**	85.22%	87.10%	88.94%
	f1	80.39%	82.22%	86.22%	85.99%	87.10%	**88.80**%
NORP	precision	89.53%	90.42%	94.13%	92.26%	92.82%	**94.76**%
	recall	88.00%	**94.61%**	91.01%	91.52%	93.42%	94.50%
	f1	88.76%	92.46%	92.55%	91.89%	93.12%	**94.63%**
PER	precision	93.73%	92.28%	90.84%	**95.83%**	95.52%	94.22%
	recall	86.74%	91.63%	95.79%	93.75%	95.21%	**96.06%**
	f1	90.10%	91.95%	93.25%	94.78%	**95.37%**	95.13%
overall	precision	88.14%	89.44%	89.30%	91.62%	92.63%	**92.91%**
	recall	84.29%	88.57%	93.42%	91.11%	92.79%	**93.72%**
	f1	86.17%	89.00%	91.32%	91.37%	92.71%	**93.32%**
	accuracy	97.09%	98.11%	97.88%	98.18%	98.42%	**98.87%**

Table 6: Performance of different models on the validation dataset.

Analysis of Glyph and Writing System Similarities using Siamese Neural Networks

Claire Roman[1], **Philippe Meyer**[2]
[1]Independent Researcher
[2]Université Paris-Saclay, INRAE, AgroParisTech, Micalis Institute, 78350, Jouy-en-Josas, France
[1]claire.roman.91@gmail.com, [2]philippe.meyer@inrae.fr

Abstract

In this paper we use siamese neural networks to compare glyphs and writing systems. These deep learning models define distance-like functions and are used to explore and visualize the space of scripts by performing multidimensional scaling and clustering analyses. From 51 historical European, Mediterranean and Middle Eastern alphabets, we use a Ward-linkage hierarchical clustering and obtain 10 clusters of scripts including three isolated writing systems. To collect the glyph database we use the Noto family fonts that encode in a standard form the Unicode character repertoire. This approach has the potential to reveal connections among scripts and civilizations and to help the deciphering of ancient scripts.

Keywords: siamese neural network, writing system, clustering

1. Introduction

The study of the comparison of scripts is interesting as it unveils links between alphabets and between glyphs, shedding light on the evolution of languages. This helps in comprehending the evolution and historical narratives of civilizations, including their migrations and interactions (Hooker, 1990; Salomon, 1998). Furthermore, it plays a pivotal role in deciphering ancient scripts and inscriptions, for example by identifying the writing systems most closely related to an undeciphered alphabet. Employing this methodological approach, Ventris and Chadwick (1953) successfully deciphered the Linear B script through a meticulous comparison with the Greek alphabet.

To apply computational linguistics and artificial intelligence to glyph comparison several issues have to be considered when choosing an appropriate model. On the one hand related graphemes could vary considerably and so a similarity function more robust to variations than usual image quality metrics such as the mean-squared error or the structural similarity (Wang et al., 2004) is needed. On the other hand artificial neural networks are widely known for their resilience to fluctuations for classification tasks (LeCun et al., 2015) but require a lot of labelled data per class which poses a challenge when comparing glyphs since thousands of classes have to be considered.

Siamese neural networks are a particular class of deep learning models that focus on discerning similarities between entries instead of classifying them into distinct categories. This makes them effective when labeled data is scarce and therefore efficient for one-shot learning (Bromley et al., 1993). They find recent applications in various fields such as intrusion detection systems (Bedi et al., 2020) or blood cell classification (Tummala and Suresh, 2023).

In this paper, we use the siamese neural networks developed by Koch et al. (2015) which have been trained and tested on the Omniglot dataset (Lake et al., 2015) in order to compare similarities between graphemes and study the space of writing systems. For that purpose we use 51 historical European, Mediterranean and Middle Eastern writing systems that we have collected from the Noto font families that encode the Unicode characters. Then we visualize the space of glyphs by multidimensional scaling analyses and we perform a Ward-linkage hierarchical clustering to define 10 families of writing systems. The dataset and codes are released at `https://github.com/PhilippeMeyer68/glyph-SNN`.

2. Related work

Various computational studies of the script evolution and comparison with the tools of mathematics, computer science and artificial intelligence have been performed. For example, families of writing systems have been obtained using clustering algorithms by minimizing the necessary topological transformations between glyphs (Hosszú and Kovács, 2016) and by using convolutional neural networks (Daggumati and Revesz, 2023). Clustering algorithms have also been used to study subgroups of a given writing system such as in Corazza et al. (2022) where unsupervised deep learning is used to classify the Cypro-Minoan writing system in one single group or in Bogacz et al. (2018) where 3D scanning and object identification are applied to visualize links between Maya glyphs.

LT4HALA 2024@LREC-COLING-2024, pages 98–104
25 May, 2024.

On the other hand deep learning models have also shown their efficiencies for glyph recognition and translation (Barucci et al., 2021, 2022; Moustafa et al., 2022; Guidi et al., 2023; Hamplová et al., 2024). In particular, Liu et al. (2022) extended the work of Koch et al. (2015) by using siamese neural networks for ancient character recognition. For an overview of published research using machine learning for ancient languages one can see the survey of Sommerschield et al. (2023).

Other approaches to decipher old scripts such as algorithms based only on non-parallel data in known languages (Luo et al., 2019) or natural phonological geometry, word segmentation and cognate alignment (Luo et al., 2021) have been conducted.

3. Method

3.1. Distances between glyphs and scripts via siamese neural networks

The model developed by Koch et al. (2015) consists of two identical convolutional neural networks that share the same set of parameters and weights. Each subnetwork takes a 105x105 pixels image as input and processes it independently through convolutional layers to generate a feature vector. These feature vectors are then compared to measure the similarity between the two input images. The network is trained using pairs of images, where one is compared to another, belonging to the same class or not, that is to say considered as positive or negative sample. A regularized cross-entropy objective loss function is employed during training to encourage the network to minimize the distance between feature vectors for images of the same class and maximize it for images of different classes. This way, the siamese network learns to extract meaningful and discriminative features that facilitate accurate similarity measurements, enabling effective one-shot learning.

To train the siamese neural network, the authors of Koch et al. (2015) use the Omniglot dataset (Lake et al., 2015) composed of 1,623 characters handwritten by 20 different individuals and from 50 alphabets, both real and invented writing systems such as the Aurebesh and Tengwar. In this work we use the same siamese neural network model, except that we train it only on the 15 invented languages of Omniglot to avoid introducing bias by comparing glyphs that would have already been used during the training phase. We still select the same random number of input pairs, that is to say 150,000 pairs of glyphs augmented with 8 distortion copies, which give 1,350,000 effective pairs.

For two glyphs g_1 and g_2 we denote by SNN(g_1, g_2) the similarity predicted by this siamese neural network and by d_g the dissimilarity measure, or distance-like function, defined by

$$d_g(g_1, g_2) := 1 - \text{SNN}(g_1, g_2). \qquad (1)$$

Let s_1 and s_2 be two scripts. We define the similarity of s_1 to s_2 by

$$\tilde{d}_s(s_1, s_2) := \frac{1}{n} \sum_{g_1 \in s_1} \min_{g_2 \in s_2} \Big(d_g(g_1, g_2) \Big), \qquad (2)$$

where n is the number of glyphs of s_1. We symmetrize it to obtain the distance-like function d_s between s_1 and s_2 defined by

$$d_s(s_1, s_2) := \frac{1}{2} \Big(\tilde{d}_s(s_1, s_2) + \tilde{d}_s(s_2, s_1) \Big). \qquad (3)$$

In this definition a glyph of s_1 can be associated with several glyphs of s_2. We believe that imposing a 1-1 mapping in the definition of d_s, such as for the bottleneck distance between persistence diagrams (Cohen-Steiner et al., 2005), is less appropriate since several glyphs can be historically related to a single glyph. For example it is known that the letters U, Y, V and W of the Latin alphabet have as ancestor the upsilon greek character Υ (Daniels and Bright, 1996).

3.2. Font-driven database

We have selected 51 historical European, Mediterranean and Middle Eastern writing systems and obtained the database of corresponding characters from their Unicode identifiers and the Noto Sans Regular family fonts.

The Unicode repertoire is an inventory of characters maintained by the Unicode Consortium and encompassing text from every writing system worldwide, facilitating global communication and interoperability across different devices and platforms.

The Noto font collection is designed and engineered for typographically correct and aesthetically pleasing global communication in more than 1,000 languages and over 160 scripts. It supports more than 77,000 characters and includes nearly all non-CJK characters included in the actual Unicode Standard version. Each supported script has at least one font in a basic style called Noto Sans Regular. This allows characters to have a standardized form, of the same size and quality.

By this process we have a database of 1,649 standardized centered glyphs as 105x105 pixels image from 51 alphabetic and syllabic writing systems. These chosen scripts are listed in Appendix A.

4. Results

4.1. Space of glyphs and scripts

In this section, we use the dissimilarity measures d_g and d_s to compare and visualize glyphs and scripts from our database. We have found that the two scripts which are the closest are the Old Sogdian and the Pahlavi Psalter with a distance of 0.05 while the most distant pair is the Coptic and the Old Persian with a distance of 0.88. Looking at how distant a script is to other writing systems by summing its distance to all other scripts we see that the Old Persian is actually the most isolated alphabet, see Table 1.

Script	Distance to other scripts
Old Persian	33.37
Glagolitic	27.69
Meroitic Hieroglyphs	26.07
Ogham	22.04
Tifinagh	21.48

Table 1: The 5 most isolated scripts with respect to the siamese-based distance.

In order to visualize graphemes and alphabets and the distances separating them we use multidimensional scaling (MDS) analysis. This is a technique in dimension reduction that aims to represent complex, high-dimensional data in a lower-dimensional space while preserving the pairwise distances between data points as accurately as possible (Kruskal, 1964).

In this way, we can represent the glyphs of one or several scripts. In Figure 1 is given the 2-dimensional scaling analysis of the Latin and Old Italic scripts, which have a distance d_s equal to 0.26. Several glyphs of these alphabets are similar and close, illustrating the real connections between these scripts, the Old Italic used in the Italian Peninsula from the 8th to the 1st century BC being known as an ancestor of the Latin, see Bonfante (1996). In Figure 2 we represent a 2-dimensional scaling analysis of the Coptic and Old Persian scripts which is the most distant pair of alphabets of the database and we notice that the alphabets essentially form two distinct clusters.

4.2. Comparison and clustering of writing systems

In this section we perform a Ward-linkage hierarchical clustering (Ward Jr., 1963) on the 51 writing systems with respect to the siamese-based distance function d_s. This agglomerative clustering algorithm analyzes the variance of the clusters and is known to be less sensitive to noise and outliers than the other hierarchical clustering algorithms.

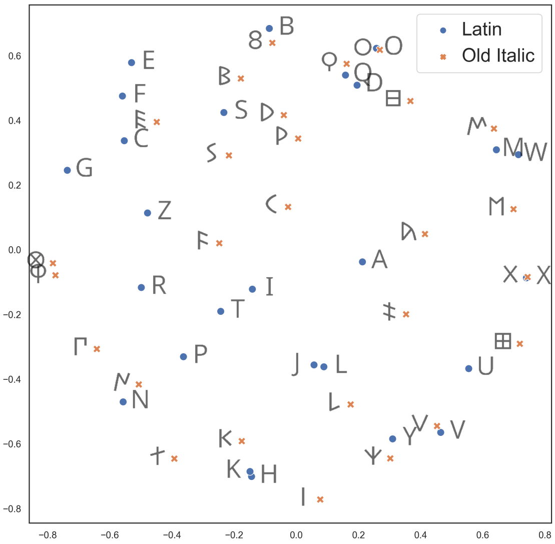


Figure 1: Multidimensional scaling in dimension 2 of the Latin and Old Italic glyphs which are close scripts with respect to the siamese-based distance.

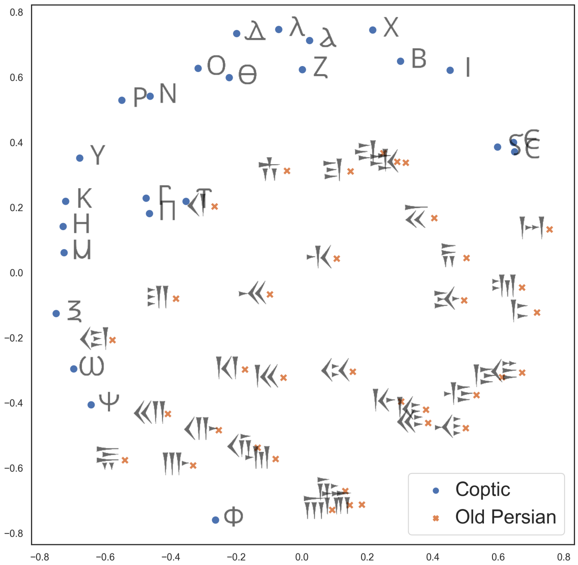


Figure 2: Multidimensional scaling in dimension 2 of the Coptic and Old Persian glyphs which are distant scripts with respect to the siamese-based distance.

The associated dendrogram of the clustering is given in Figure 3.

The Elbow method clearly indicates to truncate the dendrogram at 10 clusters. For this truncation the clustering quality Dunn index (Dunn, 1973) is 0.81. Information about size, medoid, diameter and mean distance of all pairs of each cluster is given in Table 2.

As noticed in Section 4.1, the Old Persian cuneiform, the old slavic Glagolitic and the Meroitic Hieroglyphs are isolated scripts and define their

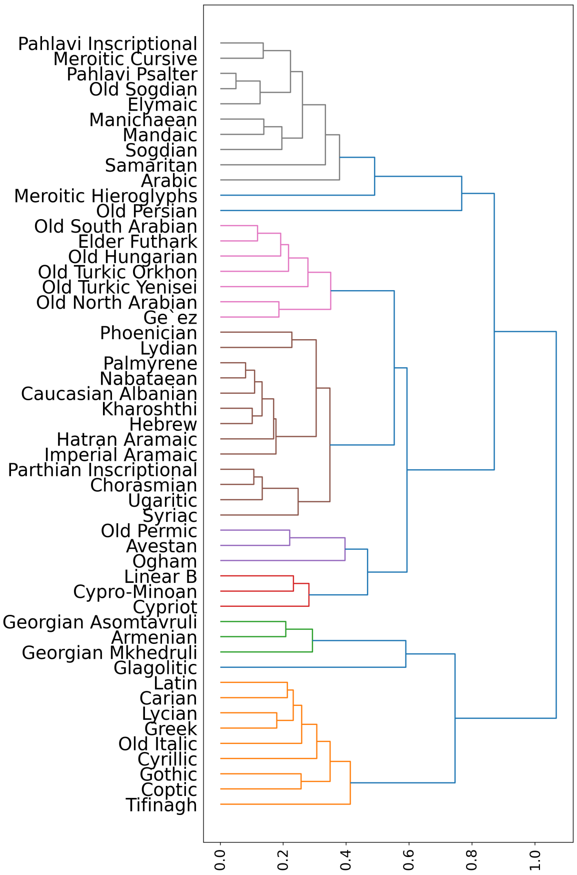


Figure 3: Dendrogram associated to a Ward-linkage hierarchical clustering of the scripts with the siamese-based distance.

Cluster	Size	Medoid	Diam.	Mean d.
C1	9	Greek	0.41	0.28
C2	3	Georgian Asomtavruli	0.28	0.25
C3	1	Glagolitic	0	0
C4	3	Cypro-Minoan	0.29	0.26
C5	3	Avestan	0.43	0.31
C6	13	Nabataean	0.40	0.19
C7	7	Old South Arabian	0.34	0.23
C8	10	Pahlavi Psalter	0.43	0.21
C9	1	Meroitic Hieroglyphs	0	0
C10	1	Old Persian	0	0

Table 2: Size, medoid, diameter and mean distance of all pairs of each cluster.

own families in the clustering. There is a cluster composed of the 3 Cypriots writing systems and another one composed of the Armenian and Georgian scripts. The three rather distant Old Permic, Avestan and Ogham scripts are grouped together. Several Middle Eastern writing systems such as the Pahlavi, the Arabic and the Sogdian form another cluster. The Greek alphabet is the medoid of a cluster composed of 9 scripts, such as the Latin or the Cyrillic and other Greek extensions such as the Carian. Another group of scripts is given of the Old Arabian and Turkic scripts. The last cluster is the biggest one, composed of Aramaic scripts that could be divided into subfamilies.

To represent how distant or close the scripts and the clusters are to each other, we perform a 2-dimensional scaling analysis and the associated visualization is given in Figure 4. We see that the distribution of the scripts respects the clusters defined by the Ward-linkage hierarchical clustering with little overlap between groups.

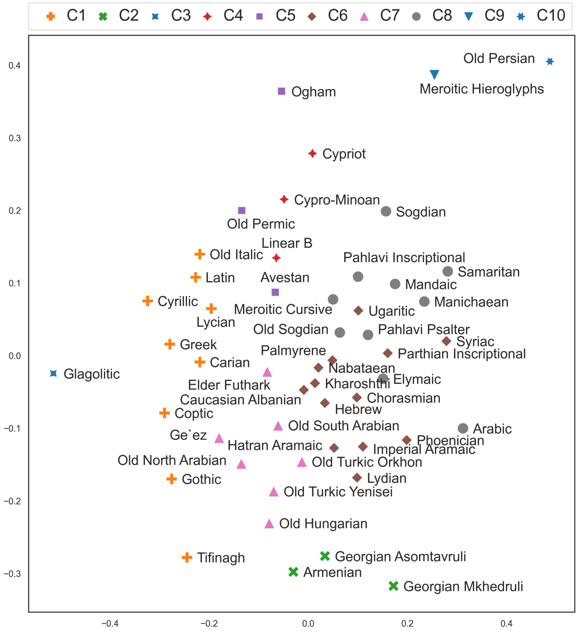


Figure 4: Multidimensional scaling in dimension 2 of all the scripts with respect to the siamese-based distance where the colors represent the 10 clusters of writing systems.

4.3. Comparison of our results with the literature

In Hosszú and Kovács (2016), 58 different historical Mediterranean and Asian scripts are classified by clustering algorithms applied on topological features of glyphs. The main similarities with our work are that there is a Latin-Greek group, a Hebrew-Nabataean group and a Cypriot group with both approaches. However, the Lydian and Phoenician scripts are in different clusters in Hosszú and Kovács (2016) while they are close with a distance of 0.23 by our metric which seems to be in agreement with the work of Woudhuizen (2020). Furthermore, the Carian script is an isolated point in Hosszú and Kovács (2016) while it is classified in the Greek family in our work. The similarities and the possible historical connection between graphemes of the Carian and the Greek scripts have been remarked and extensively studied, see Chapter 4.B *The Greek Alphabetic Era* of Adiego

(2006). Finally the Dunn index of our clustering is 0.81 which is slightly better than the Dunn index of 0.76 of Hosszú and Kovács (2016).

In the work of Daggumati and Revesz (2023), 8 ancient scripts are classified with convolutional neural networks combined with support vector machines and a hierarchical clustering. The main difference is that the Greek and the Phoenician scripts are very close with their metric whereas they are in two different clusters in our work with a siamese-based distance of 0.46. Indeed, it is known that these writing systems are related and that several glyphs of the Greek alphabet are vertical mirror reflections of Phoenician glyphs, see Swiggers (1996). It turns out that this phenomenon of boustrophedon writing is taken into account in the metric of Daggumati and Revesz (2023) but not in ours.

5. Conclusion

In this study, we introduce a two-step process for comparing glyphs and writing systems. Firstly, we present a method for generating a clean alphabet database from the Noto Sans fonts and the Unicode inventory. Then a distance-like function defined by a siamese neural network is given. This allows us to consider space of glyphs and scripts to compare them. Then a Ward-linkage hierarchical clustering of 51 alphabets resulted in the identification of 10 clusters representing related writing systems. These groups very often represent real historical connections, such as the Georgian and Armenian cluster or the Latin cluster composed of the Latin, Carian, Lycian, Greek, Old Italic, Cyrillic, Gothic, Coptic and Tifinagh scripts. This demonstrates the effectiveness of the approach in identifying links between alphabets and motivates future research to its application in deciphering ancient scripts and inscriptions.

We now discuss limitations of this approach. The comparison explained in this article is only based on the graphical aspect of the graphemes and scripts, there is no knowledge about the phonetic facet of the associated languages that intervenes. Furthermore, this work uses Unicodes and fonts and then requires an implementation of the writing systems which is not the case for all of them. For example until now there is no Unicode for the Paleohispanic scripts. Moreover, we mostly have compared segmental scripts. It is not clear if it makes sense to extend this type of comparison to logographic writing systems composed of thousands of signs such as the Chinese characters.

In future work, we would like to include all the scripts encoded in the Unicode repertoire to obtain a larger taxonomy of world's writing systems in order to contribute to the study of historical connections between civilizations. It would be particularly interesting to apply this approach to the decipherment of ancient scripts by comparing them with deciphered writing systems.

6. Bibliographical References

I. Adiego. 2006. *The Carian Language*, volume 86 of *Handbook of Oriental Studies. Section 1 The Near and Middle East*. Brill, Leiden, The Netherlands.

A. Barucci, C. Canfailla, C. Cucci, M. Forasassi, M. Franci, G. Guarducci, T. Guidi, M. Loschiavo, M. Picollo, R. Pini, L. Python, S. Valentini, and F. Argenti. 2022. Ancient egyptian hieroglyphs segmentation and classification with convolutional neural networks. In *The Future of Heritage Science and Technologies: ICT and Digital Heritage*, pages 126–139, Florence, Italy. Springer International Publishing.

A. Barucci, C. Cucci, M. Franci, M. Loschiavo, and F. Argenti. 2021. A deep learning approach to ancient egyptian hieroglyphs classification. *IEEE Access*, 9:123438–123447.

P. Bedi, N. Gupta, and V. Jindal. 2020. Siam-ids: Handling class imbalance problem in intrusion detection systems using siamese neural network. In *Third International Conference on Computing and Network Communications (CoCoNet'19)*, volume 171, pages 780–789, Trivandrum, Kerala, India.

B. Bogacz, F. Feldmann, C. Prager, and H. Mara. 2018. Visualizing networks of maya glyphs by clustering subglyphs. In *Eurographics Workshop on Graphics and Cultural Heritage*, pages 105–111, Vienna, Austria. The Eurographics Association.

L. Bonfante. 1996. The scripts of italy. In P. T. Daniels and W. Bright, editors, *The World's Writing Systems*, chapter 23, pages 297–311. Oxford University Press, Oxford, United Kingdom.

J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. 1993. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 737—-744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. 2005. Stability of persistence diagrams. In *Proceedings of the Twenty-First Annual Symposium on Computational Geometry*, SCG '05, pages

263—271, New York, NY, USA. Association for Computing Machinery.

M. Corazza, F. Tamburini, M. Valério, and S. Ferrara. 2022. Unsupervised deep learning supports reclassification of bronze age cypriot writing system. *PLOS ONE*, 17(7):e0269544.

S. Daggumati and P. Z. Revesz. 2023. Convolutional neural networks analysis reveals three possible sources of bronze age writings between greece and india. *Information*, 14(4):227.

P. T. Daniels and W. Bright. 1996. *The World's Writing Systems*. Oxford University Press, Oxford, United Kingdom.

J. C. Dunn. 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

T. Guidi, L. Python, M. Forasassi, C. Cucci, M. Franci, F. Argenti, and A. Barucci. 2023. Egyptian hieroglyphs segmentation with convolutional neural networks. *Algorithms*, 16(2):79.

A. Hamplová, A. Romach, J. Pavlíček, A. Veselý, M. Čejka, D. Franc, and S. Gordin. 2024. Cuneiform stroke recognition and vectorization in 2d images. *Digital Humanities Quarterly*, 18(1).

J. T. Hooker. 1990. *Reading the Past: Ancient Writing from Cuneiform to the Alphabet*. Barnes & Noble, Inc., New York, NY, USA.

G. Hosszú and F. Kovács. 2016. Topological analysis of ancient glyphs. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 002248–002253, Budapest, Hungary. IEEE.

G. Koch, R. Zemel, and R. Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *32nd International Conference on Machine Learning*, volume 37, Lille, France. JMLR: W&CP.

J. B. Kruskal. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Y. LeCun, Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

X. Liu, W. Gao, R. Li, Y. Xiong, X. Tang, and S. Chen. 2022. One shot ancient character recognition with siamese similarity network. *Scientific Reports*, 12(1):14820.

J. Luo, Y. Cao, and R. Barzilay. 2019. Neural decipherment via minimum-cost flow: From Ugaritic to Linear B. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3146–3155, Florence, Italy. Association for Computational Linguistics.

J. Luo, F. Hartmann, E. Santus, R. Barzilay, and Y. Cao. 2021. Deciphering Undersegmented Ancient Scripts Using Phonetic Prior. *Transactions of the Association for Computational Linguistics*, 9:69–81.

R. Moustafa, F. Hesham, S. Hussein, B. Amr, S. Refaat, N. Shorim, and T. M. Ghanim. 2022. Hieroglyphs language translator using deep learning techniques (scriba). In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 125–132, Cairo, Egypt. IEEE.

R. Salomon. 1998. *Indian Epigraphy: A Guide to the Study of Inscriptions in Sanskrit, Prakrit, and the other Indo-Aryan Languages*. Oxford University Press, Oxford, United Kingdom.

T. Sommerschield, Y. Assael, J. Pavlopoulos, V. Stefanak, A. Senior, C. Dyer, J. Bodel, J. Prag, I. Androutsopoulos, and N. de Freitas. 2023. Machine Learning for Ancient Languages: A Survey. *Computational Linguistics*, 49(3):703–747.

P. Swiggers. 1996. Transmission of the phoenician script to the west. In P. T. Daniels and W. Bright, editors, *The World's Writing Systems*, chapter 21, pages 261–270. Oxford University Press, Oxford, United Kingdom.

S. Tummala and A. K. Suresh. 2023. Few-shot learning using explainable siamese twin network for the automated classification of blood cells. *Medical & Biological Engineering & Computing*, 61:1549—-1563.

M. Ventris and J. Chadwick. 1953. Evidence for greek dialect in the mycenaean archives. *The Journal of Hellenic Studies*, 73:84–103.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

J. H. Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

F. C. Woudhuizen. 2020. The lydian yod-sign. In Gür B. and Dalkılıç S., editors, *A Life Dedicated to Anatolian Prehistory. Festschrift for Jak Yakar*, chapter 32, pages 465–477. Bilgin Kültür Sanat, Ankara, Turkey.

A. List of scripts

To collect the glyph database we have selected all the European, Mediterranean and Middle Eastern writing systems that are implemented in the version 15.0 of the Unicode Standard, see Table 3. Many of these writing systems are alphabetic such as the Latin and Lycian scripts while some of them are abjad, abugida or syllabic writing systems such as the Arabic, Cypriot and Ge`ez scripts (Daniels and Bright, 1996).

Script	Number of glyphs
Arabic	36
Armenian	38
Avestan	54
Carian	49
Caucasian Albanian	52
Chorasmian	21
Coptic	25
Cypriot	55
Cypro-Minoan	97
Cyrillic	32
Elder Futhark	24
Elymaic	22
Ge`ez	26
Georgian Asomtavruli	38
Georgian Mkhedruli	33
Glagolitic	47
Gothic	27
Greek	24
Hatran Aramaic	21
Hebrew	27
Imperial Aramaic	22
Kharoshthi	37
Latin	26
Linear B	60
Lycian	29
Lydian	26
Mandaic	25
Manichaean	36
Meroitic Cursive	24
Meroitic Hieroglyphs	30
Nabataean	31
Ogham	20
Old Hungarian	51
Old Italic	27
Old North Arabian	29
Old Permic	38
Old Persian	36
Old Sogdian	18
Old South Arabian	29
Old Turkic Orkhon	42
Old Turkic Yenisei	31
Pahlavi Inscriptional	19
Pahlavi Psalter	18
Palmyrene	23
Parthian Inscriptional	22
Phoenician	22
Samaritan	22
Sogdian	21
Syriac	26
Tifinagh	31
Ugaritic	30

Table 3: The writing systems used in this work.

How to Annotate Emotions in Historical Italian Novels: a Case Study on *I Promessi Sposi*

Rachele Sprugnoli, Arianna Redaelli
Università di Parma, Viale D'Azeglio, 85, 43125 Parma, Italy
{rachele.sprugnoli, arianna.redaelli}@unipr.it

Abstract

This paper describes the annotation of a chapter taken from *I Promessi Sposi*, the most famous Italian novel of the 19th century written by Alessandro Manzoni, following 3 emotion classifications. The aim of this methodological paper is to understand: i) how the annotation procedure changes depending on the granularity of the classification, ii) how the different granularities impact the inter-annotator agreement, iii) which granularity allows good coverage of emotions, iv) if the chosen classifications are missing emotions that are important for historical literary texts. The opinion of non-experts is integrated in the present study through an online questionnaire. In addition, preliminary experiments are carried out using the new dataset as a test set to evaluate the performances of different approaches for emotion polarity detection and emotion classification respectively. Annotated data are released both as aggregated gold standard and with non-aggregated labels (that is labels before reconciliation between annotators) so to align with the perspectivist approach, that is an established practice in the Humanities and, more recently, also in NLP.

Keywords: annotation, emotion analysis, historical texts, Italian literature, 19th century Italian

1. Introduction

Emotion analysis is a task at the intersection of Natural Language Processing (NLP) and Affective Computing whose aim is to automatically recognize the emotions conveyed in a text. It is important to note that the concept of emotion is notoriously difficult to define (Scherer, 1984); for the purposes of this paper, we will use the word "emotion" as an umbrella term to encompass various affective states including all kinds of feelings, moods, attitudes, and behavioral responses.

Applications, domains and text genres considered in the emotion analysis task are extremely varied (Acheampong et al., 2020) and the organization of specific evaluation exercises in various languages demonstrates the growing interest of the NLP community towards the analysis of emotions (Mohammad et al., 2018; Plaza-del Arco et al., 2021; Araque et al., 2023). In this context, literary texts are less studied in NLP than, for example, social media posts but, on the contrary, the relationship between emotions and literary texts is of enormous interest in the field of Digital Humanities especially after the so-called affective-turn in literary studies (Keen, 2011). Therefore, emotion analysis is a task where a collaboration between the two communities can be extremely fruitful and beneficial for both. This paper[1] presents an example of such collaboration by describing the sentence-level emotion annotation of a chapter from a 19th century novel (for a total of 338 sentences and more than 9,000 tokens) according to 3 distinct classifications. The purpose of this paper is mostly methodological; instead of aiming for a large amount of data, in this phase we want to study in depth: i) how the annotation changes depending on the granularity of the classification, ii) how the different granularities impact the inter-annotator agreement, iii) which granularity allows good coverage of emotions, and iv) if the chosen classifications are missing emotions that are important for a literary text of the 19th century. To achieve these goals, a questionnaire was also created involving 45 anonymous non-experts.

The data of our study are from the final edition (1840-1842) of Alessandro Manzoni's *I Promessi Sposi* (*The Betrothed*). This novel is fundamental to both the history of Italian literature and the development of the Italian language, as it introduced a functional model of written literary language that closely mirrored common speech and was widely imitated by Italian authors, scholars and learners. Following the unification of Italy in 1861, the novel emerged as a symbol of national identity, and its prominence was particularly felt in the educational sector, where it was swiftly incorporated into the literary canon. This not only strengthened its status as a cornerstone of Italian literature but also positioned it as a practical model from which to learn Italian language and even derive grammatical norms to be taught in schools. However, over the years, this educational emphasis cast the novel in a somewhat gray, heavy, and static light for many students. This per-

[1] This paper is the result of the collaboration between the two authors. For the specific concerns of the Italian academic attribution system: Rachele Sprugnoli is responsible for Sections 2, 4, 5 and 6; Arianna Redaelli is responsible for Sections 1 and 3. Section 7 was collaboratively written by both authors.

ception stands in stark contrast to the novel's true nature, which is dynamic and original. Furthermore, Manzoni's meticulous exploration of the language of passions, underscored by a moral perspective (Maiolini et al., 2017), ensures the novel's emotional depth and variety. Such qualities, together with the intricate narrative and well-rounded characters, far from melodramatic stereotypes, not only affirm its status as a literary masterpiece, but also highlight its suitability for emotion analysis. In turn, emotion analysis can even serve as a mean to reemphasize the novel's positive features, potentially revitalizing its perception in education and encouraging renewed appreciation among students.

From the data availability standpoint, the text of *I Promessi Sposi* is free from copyright, fully digitized, and available in a machine-readable and clean (that is without OCR errors) format. This format ensures seamless integration with computational tools with minimal manual intervention.

To sum up, our main contributions are as follows: i) an in-depth study on the annotation of emotions in an Italian historical literary text that, despite its critical significance to Italian literary history, has not previously been examined through NLP methods; ii) the development of a new dataset manually annotated with 3 emotion classifications of different granularity that is released with both aggregated and non-aggregated annotations; iii) the release of a new polarity lexicon derived from 19th-century Italian narrative texts.[2]

2. Related Work

Over the last few years, numerous datasets for emotion analysis have been developed following two main approaches. The first approach is based on the idea that emotions are innate, universal and limited in number, thus they can be classified using categorical labels, often borrowed from psychological theories, such as those of Ekman (Ekman, 1992) and Plutchik (Plutchik, 1980). On the contrary, in the second approach, emotions are represented by combining a small set of dimensions using continuous values. For example, Russell and Mehrabian (1974) identify valence (degree of pleasantness), arousal (degree of excitement) and dominance (degree to which a person feels in control of a situation) as the three fundamental dimensions for defining all emotions. From this theory derives the so-called VAD (Valence-Arousal-Dominance) model which serves as the foundation for both lexicons and annotated datasets, see among others (Buechel and Hahn, 2017; Mohammad, 2018). Both approaches have advantages and disadvantages: categorical classifications are intuitive to understand but use culture- and language-specific labels that are not actually universal, while dimensional models can describe feelings that would otherwise be difficult to label but are harder to interpret by humans. Therefore there are studies that aim not only to analyze the two approaches but also to unify them (Calvo and Mac Kim, 2013; Bostan and Klinger, 2018). However, in our work we have decided to adopt a discrete classification for its ease of interpretation because, as anticipated in Section 1, our aim is to create a resource easily accessible even to non-experts, humanities scholars and students first and foremost.

The main issue when dealing with the categorical approach is the choice of the classification to adopt. Together with works that borrow Ekman's 6 emotions[3] or Plutchik's 8 basic emotions[4], usually adding a label for neutral cases (Alm et al., 2005; Schuff et al., 2017; Öhman et al., 2020), there are also datasets that employ a much narrower or much broader set. For example, Grounded-Emotions is annotated only with `sadness` and `happiness` (Liu et al., 2017), while FEEL-IT with `anger`, `fear`, `sadness` and `joy` (Bianchi et al., 2021). On the contrary, the dataset of SemEval-2018 Task "Affect in Tweets" uses 11 emotions[5] (Mohammad et al., 2018) and Demszky et al. (Demszky et al., 2020) propose a taxonomy of 27 categories plus `neutral` (see Section 3.2 for the complete list). Although the various classifications are often applied to texts that are very different from each other (e.g., posts on social media, song lyrics, transcriptions of dialogues) in an indistinct manner, some works instead focus on how to find the most suitable taxonomy for the textual genre to be annotated. This is particularly important for literary texts where emotions tend to be complex, subtle and intertwined with narrative, aesthetic and cultural aspects. For example, for the annotation of historical German plays different annotation schemes have been tested (Schmidt et al., 2018), and then 13 hierarchically structured emotion concepts have been defined (Schmidt et al., 2021). On the other hand, in the Kāvi corpus, Punjabi poems are annotated following the concept of Navrasa, that distinguishes nine emotions, such as "shaanti" (meaning *peace*) and "raudra" (meaning *anger*), in order to better reflect Indian culture (Saini and Kaur, 2020). The survey papers by Kim and Klinger (2019) and Reb-

[2] All data presented in this paper are available in a GitHub repository: `https://github.com/RacheleSprugnoli/Emotion_Analysis_Manzoni`

[3] `Anger, disgust, fear, joy, sadness, surprise.`

[4] `Anger, disgust, fear, joy, sadness, surprise, trust, anticipation.`

[5] `Anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, and trust.`

ora (2023), to which we refer for further details, well describe the broad and multifaceted panorama of emotion and sentiment analysis applications in the field of computational literary studies.

In the present work we decided not to uncritically adopt one classification but to try different taxonomies to identify the one that best suits our case study and that can be potentially applicable to other Italian novels as well. Furthermore, the annotated data produced in this work enriches the inventory of linguistic resources for emotion analysis available for Italian which, although always growing, is not as abundant as for other languages. Notable examples of recent Italian datasets in the field of emotion analysis are: FEEL-IT (tweets annotated with 4 emotions, see above), the EMit dataset (Araque et al., 2023) (tweets annotated with Plutchik's basic emotions plus `love` and `neutral`), MultiEmotions-it (Sprugnoli, 2020) (comments posted on YouTube and Facebook annotated with both Plutchik's basic and complex emotions) and AriEmozione (Zhang et al., 2022) (opera verses annotated with 6 emotions, namely, `love, joy, admiration, anger, sadness` and `fear`).

3. Data and Annotation

This Section describes the data used in our annotation and the workflow we followed giving details on the selected chapter and on the emotion taxonomies adopted.

3.1. Data Selection

Among the 38 chapters of the novel, chapter VIII appeared to be the most suitable one to start the annotation. Indeed, this chapter is particularly noteworthy for its structure, consisting of 5 macro-sequences: the failed marriage attempt in the house of the priest Don Abbondio, the failed kidnapping of Lucia (the female protagonist) by the *bravi* (hired assassins), the gathering of the crowd outside Don Abbondio's house at the tolling of the bell, the meeting of the bethroted and Lucia's mother (Agnese) with Fra Cristoforo (a monk) in a church, and the abandonment of the hometown. Given the profound diversity of the aforementioned themes, chapter VIII also shows a wide range of scenes and tones (moving between the extremes of Don Abbondio's sympathetic opening line and Lucia's final weeping), and a great stylistic-narrative variety (shifting from dialogue to vivid description, and finally to the lyrical depth of the *Addio ai Monti* [*Farewell to the mountains*]). Additionally, the many events of the chapter involve more than 15 characters, each one distinctly marked by his own linguistic features, gestures, and emotional states. Furthermore, chapter VIII is one of the longest in the novel (9.808 tokens, including punctuation) which allowed us to have a good amount of textual material to annotate.

negative	0.78
neutral	0.76
positive	0.57
mixed	0.46
overall	0.73

sadness	0.79
fear	0.75
anger	0.73
surprise	0.72
joy	0.69
neutral	0.57
anticipation	0.53
trust	0.53
disgust	0.44
overall	0.53

Table 1: Inter-annotator agreement in terms of Krippendorff's Alpha for emotion polarity annotation (on the left) and for the annotation of Plutchik's basic emotions (on the right).

For all these reasons, chapter VIII not only offers a microcosm of the novel's intricate emotional and linguistic features but also provides a comprehensive and varied dataset for emotion analysis. By focusing on this chapter, we were allowed to obtain a condensed and yet diverse representation of the emotional dynamics that permeate the entire novel of *I Promessi Sposi*.

3.2. Annotation Workflow

The annotation was carried out using a simple spreadsheet with a sentence per line in their original order.[6] Sentence splitting was performed manually because the automatic segmentation proved to be very challenging for the models currently available for Italian due to issues related to the novel's complex punctuation. For example, the text contains low quotation marks («») indicating direct speech spoken aloud, while the long dash (–) is used to delimit thought or muttered direct speech. These punctuation marks, and their so specific and diverse use, are not common in contemporary texts, thus systems are not trained to recognize them correctly. For example, an accuracy of 64% was registered with Stanza (Qi et al., 2020). At the end of the manual sentence splitting procedure, we obtained 338 sentences of different length (from 1 to 109 tokens).

Two annotators were involved in the annotation: one with a significant expertise in Manzoni's work but limited annotation experience, and the other being an experienced annotator with basic knowledge of Manzoni. The first 20 sentences were annotated collaboratively, while the remaining sentences were

[6] By "sentence" we mean a coherent set of words that conveys a complete thought and ends with a strong punctuation mark (e.g., full stop, question mark, or exclamation point), typically followed by a capital letter.

love	0.86
curiosity	0.83
sadness	0.75
gratitude	0.74
fear	0.73
anger	0.71
neutral	0.71
disgust	0.67

remorse	0.66
optimism	0.66
nervousness	0.65
embarrassment	0.59
joy	0.57
disapproval	0.55
surprise	0.45
confusion	0.42

annoyance	0.42
admiration	0.33
relief	0.30
caring	0.29
disappointment	0.15
approval	NEG
desire	NEG
realization	NEG
overall	0.44

Table 2: Inter-annotator agreement in terms of Krippendorff's Alpha for the annotation using GoEmotions classification.

annotated independently by each annotator following 3 types of emotion classification.

The first classification takes into consideration the polarity of the emotions conveyed by the text. More specifically, emotion polarity is categorized into 4 classes: i) positive (meaning that positive emotions are clearly prevalent in the sentence), ii) negative (which means that negative emotions are clearly prevalent in the sentence), iii) mixed (which indicates that opposite emotions are expressed in the sentence and it is not possible to find a clearly prevailing emotion polarity), iv) neutral (to be used when no emotions are expressed in the sentence). This coarse-grained taxonomy requires a single-label annotation while the other two adopted classifications allow a multi-label annotation being Plutchik's basic emotions and the taxonomy proposed for the GoEmotions dataset (Demszky et al., 2020). The first consists of 8 labels (namely, anger, fear, sadness, disgust, surprise, anticipation, trust, and joy) plus neutral, whereas the second is made of 27 distinct emotion categories (admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, and surprise) plus neutral.

The guidelines prescribed, for all 3 annotation types, to: i) evaluate both the lexicon used and the images evoked (for example through the use of rhetorical figures) in the sentence, ii) focus on the emotions expressed by the author, either directly (as the narrator present in the story) or indirectly (through the characters), and not on those perceived by the reader; iii) take into consideration the flow of the narrative also considering the previous sentences but not the ones that follow. Subsequently, for each classification, the individual labels were explained; for example, for the GoEmotions taxonomy the brief descriptions reported in the corresponding paper were taken (Demszky et al., 2020).

neutral	166
negative	129
mixed	22
positive	21

neutral	133
anticipation	75
fear	68
anger	52
surprise	29
sadness	25
trust	24
joy	11
disgust	5

Table 3: Number of annotated labels after reconciliation: emotion polarity on the left and Plutchik's basic emotions on the right.

4. Data Analysis

This section presents details on the inter-annotator agreement (IAA) and on the dataset obtained after the reconciliation of disagreements.

4.1. Inter-Annotator Agreement

Tables 1 and 2 report the results of the IAA in terms of Krippendorff's alpha for each label and for each classification together with the overall score. Labels are ranked in descending order of agreement. The overall scores show a substantial agreement for emotion polarity annotation (0.73) and a moderate agreement for both the annotation of Plutchik's basic emotions (0.53) and the GoEmotions classification (0.44). Given the well-known high subjectivity of emotion annotation and the multi-label nature of two of the three used classifications, these results can be considered promising.

The IAA on single labels varies greatly: such wide variability is common in emotion annotation, as attested in several previous works, for example (Strapparava and Mihalcea, 2008; Schuff et al., 2017). In the emotion polarity annotation, the negative and neutral classes proved to be the easiest to annotate (0.78 and 0.76, respectively), followed by positive (0.57), whereas mixed was the most problematic (0.46). Although difficult to recognize, we think that the mixed class is important because it captures the complexity of the literary text. Eliminating that class would impoverish the annotation making it less interesting for humanities scholars. Among the Plutchik's basic emotions, the highest scores were achieved with three negative emotions (sadness, fear, anger) however, even in this case, the agreement is between substantial and moderate for all the labels. Moreover, 64% of the sentences have both the annotators agreeing on at least one emotion label. As for the GoEmotions taxonomy, 17 labels out of 24 have at least a moderate agreement but approval, desire and realization registered slightly negative values (-0,004, -0,007 and -0,007 respectively) indicating an inverse agreement, less than that expected by chance. Indeed, these 3 classes had been misinterpreted by an annotator who had never used them. However, in general, we note that the values are on average higher than those reported for the original English dataset. In addition, 81% of the sentences have the annotators agreeing on at least one emotion label.

4.2. Annotated Data after Consolidation

Disagreements were discussed and consolidated to obtain gold labels. Our consensus-building efforts was primarily centered on enhancing the annotation methodology itself, enabling us to adjust our guidelines and labels for clearer future annotations. Using Plutchik's and GoEmotions classifications, most of the sentences resulted with a single emotion label (77% for the former and 64% for the latter, respectively), followed by sentences with 2 labels (22% and 33%, respectively) while 3 emotions are a strong minority (1,5% and 3%, respectively).

Tables 3 and 4 present the number of labels for each classification after the reconciliation in descending order. The neutral class is always the most frequent: it makes up 49% of all the labels in the emotion polarity annotation, 31% in Plutchik's classification and 24% in the GoEmotions annotation. The fact that the number of neutral sentences is not constant is due to the greater annotation granularity allowed by the Plutchick's and GoEmotions classifications. Having much more detailed labels available, led annotators to be able to better specify emotional nuances, recognizing them more easily. In particular, what is annotated as neutral in the first classification is instead marked with an ambiguous emotion (namely, surprise and anticipation following the Plutchick's distinction, realization, surprise, curiosity, confusion in GoEmotions) in the others. For example, the first sentence of the chapter (the exclamation of a proper name), i.e., "– Carneade!" (EN: - *Carneades!*), is annotated as neutral, surprise, surprise respectively. Often, a sentence marked as neutral in the emotion polarity annotation is marked as anticipation following the Plutchik's annotation and as nervousness following GoEmotions taxonomy: this last label makes explicit the anxiety that underlies the expectation of an event disambiguating an ambiguous emotion. An example is given by the sentence "Entraron pian piano, in punta di piedi, rattenendo il respiro; e si nascosero dietro i due fratelli."[7]

Apart from the neutral class, there is a large disparity in terms of label frequency. Although a similar disparity is also present in the GoEmotions dataset, a very different distribution of emotions is noted due to the different nature of the texts considered. In fact, the most frequent labels in the English GoEmotions data are admiration and approval whereas in Manzoni's chapter negative and ambiguous emotions prevail. It is interesting to note that the strong presence of negative emotions in our data is also attested in other literary datasets, such as (Zhang et al., 2022) and (Schmidt et al., 2021), regardless the annotation scheme used.

To better understand the relationship between emotions across the three types of annotation, we calculated the correlation between emotion polarities and the classes of Plutchik and GoEmotions. More specifically, we converted emotion labels into their corresponding polarity value leaving out ambiguous emotions. For example, anger and embarrassment were mapped onto the negative class, whereas joy and approval onto the positive one. Annotations made of opposite emotions (as the third sentence in Table 7) were converted into the mixed class. We found a strong positive correlation both between the emotion polarity annotation and the Plutchik's classification (0.70) and between the emotion polarity annotation and the GoEmotions classification (0.76).

5. Preliminary Experiments

The small size of the dataset did not allow it to be used to train new models but was instead adopted as a test set. In particular, we tried two approaches for polarity detection:

- Lexicon-based: a score is computed for each sentence by summing the polarity values of the tokens as recorded in a polarity lexicon (see below for more details). Positive and negative labels are assigned to sentences with a score above or below zero, respectively. Instead, we assign the neutral label to sentences in which all words have a score of 0 and the mixed label when the positive and negative values balance each other resulting in a sum of 0.

[7] EN: *They came in slowly slowly, on tiptoe, holding their breath, and hid behind the two brothers.* (Manzoni, 2022)

neutral	111
nervousness	74
fear	42
curiosity	33
disapproval	25
anger	23
surprise	23
confusion	22
caring	21
annoyance	15
relief	13
sadness	13
optimism	10
disappointment	7
desire	5
embarrassment	5
gratitude	5
remorse	5
joy	4
love	4
approval	3
admiration	2
disgust	2
realization	2

Table 4: Number of annotated labels after reconciliation for the annotation using GoEmotions classification.

LEXICON-BASED: W-MAL			
	P	R	F1
pos	0.08	0.76	0.14
neg	0.57	0.54	0.56
neu	0.85	0.07	0.12
mix	0.00	0.00	0.00
avg	**0.37**	**0.34**	**0.21**

LEXICON-BASED: XIX Cent.			
	P	R	F1
pos	0.17	0.67	0.27
neg	0.67	0.56	0.61
neu	0.78	0.55	0.64
mix	0.22	0.32	0.26
avg	**0.46**	**0.52**	**0.45**

CROSS-LANGUAGE SYSTEM			
	P	R	F1
pos	0.22	0.67	0.33
neg	0.65	0.37	0.47
neu	0.63	0.75	0.68
mix	0	0	0
avg	**0.37**	**0.45**	**0.37**

Table 5: Results of emotion polarity detection in terms of precision (P), recall (R) and F1-measure (F1).

- Cross-lingual model: a zero-shot cross-language system (Sprugnoli et al., 2023) that classifies emotion polarity into the same 4 classes used in our annotation, trained on an English dataset of social media texts and fine-tuned on XLM-RoBERTa (Conneau et al., 2020).

As for emotion classification, we tested two off-the-shelves models:

- FEEL-IT (Bianchi et al., 2021): a monolingual emotion classification system, trained on Italian tweets, that identifies 4 emotions (`fear`, `joy`, `sadness`, `anger`). We evaluated this tool only on the 73 sentences annotated with these emotions.

- XLM-EMO (Bianchi et al., 2022): a multilingual emotion classification system, fine-tuned on XLM-RoBERTa, that identifies the same emotions as FEEL-IT. Also in this case, only 73 sentences were used for the evaluation being them annotated with `fear`, `joy`, `sadness` or `anger`.

For emotion polarity detection, the lexicon-based approach relied on two polarity lexicons. The first one, W-MAL (Vassallo et al., 2020), is based on contemporary Italian whereas the second was developed to be more representative of the lexical characteristics of 19th century Italian. For this reason, we downloaded[8] the narrative texts published in the period of interest (including *I Promessi Sposi*), listed the tokens in order of frequency and assigned a polarity value (i.e. -1 for negative polarity, +1 for positive polarity and 0 for neutral cases) to all the tokens with a frequency higher or equal to 5. The final lexicon is made of 18,885 entries with a strong majority of neutral tokens (69.1% of the total) and more negative entries (19.5% of the total) than positive ones (11.4% of the total). The IAA calculated on a randomly chosen subgroup consisting of 10% of the entries was substantial (Cohen's kappa = 0.76).

As reported in Table 5, the lexicon-based approach using this new lexicon achieved the best F1 (0.45, weighted macro-average F1 0.58, accuracy 54) and it is the only method capable of identifying sentences with mixed polarity, even if only 7 times out of 22. Performances on the `neutral` and `negative` classes are good but, on the contrary, they are low on `positive`. A similar pattern is registered for the cross-lingual model,[9] whereas with the W-MAL lexicon a good F1 is achieved only for the `negative` class.

As for emotion classification, Table 6 shows that the multilingual model performed better than the monolingual one obtaining a F1 of 0.47 (weighted macro-average F1 0.50, accuracy 0.49). However, precision and recall are non balanced, with the latter being higher than the former. For FEEL-IT the lowest performance was on `fear`, which is the least frequent emotion in the training corpus and the most difficult to recognize even in the experiments carried out by the system developers. Instead, in the case of XLM-EMO the lowest F1 was registered for `joy` for which the recall is perfect but the precision is very low.

These results confirm the need to create adequate

[8] `http://www.bibliotecaitaliana.it/`.

[9] Please note that these results are worse than those that the same system obtained both on Italian social media texts and on Opera verses written in 18th-century Italian.

FEEL-IT			
	P	R	F1
anger	0.62	0.56	0.59
fear	0.38	0.11	0.17
sadness	0.29	0.67	0.41
joy	0.14	0.33	0.20
macro avg	**0.36**	**0.42**	**0.34**

XLM-EMO			
	P	R	F1
anger	0.54	0.52	0.53
fear	0.71	0.36	0.48
sadness	0.50	0.60	0.55
joy	0.20	1.00	0.33
macro avg	**0.49**	**0.62**	**0.47**

Table 6: Results of emotion classification in terms of precision (P), recall (R) and F1-measure (F1).

resources for the development of new models suitable for the processing of historical literary texts.

6. Emotion Annotation Elicitation

An additional study involved non-experts through an online questionnaire (made with Google Form) circulated on social networks (namely, LinkedIn, Mastodon and X). We selected 21 sentences taken from chapter VIII (i.e., the same text annotated by experts). These sentences belong to three textual passages chosen for their structural and emotional differences in order to present a good variability without, however, making the questionnaire too long (consequently reducing the risk of non-completion by the participants). The first group of sentences describes the final agitated phases of the failed attempt at marriage between Renzo and Lucia; the second is a sequence of short direct speeches between the crowd who rushed to help Don Abbondio and the priest himself, who regretted having raised the alarm; the third passage reports Lucia's thoughts while, on board a boat, she sadly says goodbye to her beloved homeland. Instructions were as follows.[10] "*Your task is to tell us which emotions you think are expressed in each sentence. For each sentence you can report one or more emotions; we won't give you a list of emotions to choose from, but you can express yourself freely. The sentences are taken from chapter VIII of The Betrothed by Alessandro Manzoni (1840). Read one sentence at a time and indicate the emotions that you think are expressed and/or felt by the narrator or the characters. ATTENTION: not what you feel when reading the sentence. If you want to list multiple emotions, separate them with a comma; if you can't express the emotion with a single word, also describe it with a sentence or a phrase; if it seems to you that the text does not express any emotion, write NO.*" Under the instructions, the groups of sentences were presented in distinct sections so as to make it clear that they were separate units. We also collected some socio-demographic information: namely, age (i.e., under 18, between 18 and 29, between 30 and 50, over 60), self-perceived gender identity (i.e., male, female, other, I prefer not to specify) and level of education (i.e., high school diploma, bachelor's degree, master's degree, Phd).

In one week we collected 45 responses. In general, the most mentioned emotions for each sentence correspond to those identified by the experts (see Table 7)[11] but, not having given a predefined list of labels, we recorded a great lexical richness with the use of numerous synonyms and plesionyms. For example, *spavento* (fright), *timore* (dread), *angoscia* (anguish), *panico* (panic), *terrore* (terror), *orrore* (horror), *allarme* (alarm), *sgomento* (dismay) can be traced back to the `fear` label, while `anger` is expressed also with words such as *furia* (fury), *collera* (wrath), *ira* (rage), *odio* (hate), *aggressività* (aggression). This observation prompted us to enhance the guidelines by incorporating lists of synonyms into the descriptions of emotions, thereby clarifying that each label encompasses a range of emotional shades.

The analysis of the responses also highlighted the recurring emergence of some emotions, such as `resignation`, not present in the classifications used by experts; adding such labels could make the annotation more precise but their adoption must be carefully evaluated to avoid that the increase in labels leads to a decrease in agreement.

Finally, the responses were analyzed from the point of view of the socio-demographic characteristics of the participants. In particular, we studied the propensity to assign more than one emotion per sentence based on differences in age, gender and education level. The only statistically significant difference detected (with alpha = 0.05) is the one between males and females, with the latter indicating more emotions per sentence than the former.

[10] The original instructions were written in Italian.

[11] Translations of sentences in Table 7, taken from (Manzoni, 2022): i) *Having dropped the lamp he'd been holding, he used that hand to gag her with the cloth, almost suffocating her. And all the while he kept shouting at the top of his lungs, "Perpetua! Perpetua! Treachery! Help!"*; ii)*And saying this, he stepped back and closed the window once more.*; iii) *Farewell childhood home, where lost in private thoughts, she had learned to hear the difference between normal footsteps and the footsteps of the youth she awaited with a mysterious fear.*

Sentence	Polarity	Basic	GoEmotions	Questionnaire
E subito, lasciata cader la lucerna che teneva nell'altra mano, s'aiutò anche con quella a imbacuccarla col tappeto, che quasi la soffogava; e intanto gridava quanto n'aveva in canna: «Perpetua! Perpetua! tradimento! aiuto!»	negative	anger,fear	anger,confusion,fear	fear,anger
E, detto questo, si ritirò, e chiuse la finestra.	neutral	neutral	neutral	no
Addio, casa natìa, dove, sedendo, con un pensiero occulto, s'imparò a distinguere dal rumore de' passi comuni il rumore d'un passo aspettato con un misterioso timore.	mixed	sadness	love,sadness	nostalgia,sadness

Table 7: Examples taken from our data after reconciliation; the last column presents the two most mentioned emotions in the questionnaire. Please note that the answers to the questionnaire are translated into English from the original Italian. Sentence translation is provided in footnote 11.

7. Discussion and Conclusion

This paper describes, from a methodological point of view, the annotation of emotions in a chapter taken from *I Promessi Sposi*, the most famous Italian novel of the 19th century written by Alessandro Manzoni. The annotation was based on 3 different classifications with the final goal of finding the best taxonomy for a historical literary text, balancing the richness of the recognized emotional states and the feasibility of the annotation. During their work, annotators could add any suggestions or doubt in an ad-hoc field. Other useful suggestions came from a questionnaire, aimed at non-experts, that helped us improve the guidelines. The following issues emerge from the comments by both annotators and non-expert. Regarding emotion polarity, annotators felt the lack of a label to indicate sentences with ambiguous emotions. On the other hand, Plutchik's emotions were not always considered suitable because they were too generic: indeed, often the annotator chose an emotion going solely by exclusion (a repeated comment was "it seems to me that none of the other options are suitable"). Finally, the lack of a label to indicate resignation is reported when annotating with the GoEmotions taxonomy: this need was declared also by non-experts (see Section 6). An additional suggestion was to introduce a specific level of annotation for irony. This proposed layer aims to address the subtle use of irony in Manzoni's writing, a topic extensively analyzed by literary critics (see (Raimondi, 1990; Mancini, 2005) among others), and its correlation with the emotional features of the text. The feasibility of integrating this layer and its impact on inter-annotator agreement are subjects for further investigation.

In addition, preliminary experiments were carried out using the new dataset as a test set to evaluate off-the-shelves tools for emotion polarity detection and emotion classification respectively. In this context we developed a new lexicon created by assigning a polarity value to almost 19,000 tokens taken from 19th-century Italian narrative texts. The outcomes of the experiments underscore the necessity for developing systems tailored to process historical literary texts, which have very specific linguistic features.

Going back to the aims of the work listed in Section 1, we can summarize the results obtained by our study as follows. The presence of more detailed labels leads to a wider recognition of the different emotional nuances and a reduction in the number of neutral sentences. As expected, a greater granularity of the classifications is accompanied by a lower agreement: however, the majority of emotions have an IAA between substantial and moderate. The 27 classes of the GoEmotions taxonomy seem to be suitable for representing the complexity of the literary text but it is necessary to add a label to express resignation, and to refine the guidelines by adding more information to each emotion description (for example, providing a list of synonyms and plesionyms).

We release the annotated data both with aggregated and non-aggregated labels. The annotation elicited from the questionnaire are also available. Offering more than one perspective in identifying emotions allows this study to be in line with established practices in the Humanities. For example, a central assumption of contemporary literary theory is that facts, values, reason, and nature are constructs, not objective and immutable realities (Fischer, 1990). According to this theory, literary texts are not static entities but are open to multiple interpretations, each shaped by the unique perspective of the reader or critic. This notion suggests that a single text can offer a multitude of readings, each valid in its own right, and emphasizes the importance of understanding literature as a dynamic interplay between text and reader. This approach aligns perfectly with the perspectivist turn in the

field of NLP (Cabitza et al., 2023), which we see as an interesting topic for future collaboration between NLP and DH scholars.
Future work extends in at least three directions. First, we want to expand the annotation to other chapters so that we have more data and can run new experiments and train new models. Secondly, we plan to apply the same type of annotation to other Italian novels to verify the degree of generalization of the proposed approach. Another interesting future study concerns the annotation of the emotions as elicited in the reader that would allow us to have two complementary points of view on the same text. This double approach (writer- and reader- oriented) is particularly suitable for literary texts, as demonstrated by recent reader response studies (Rebora, 2023; Pianzola et al., 2020), and further highlights the need to include multiple perspectives in the computational analysis of emotions.

8. Acknowledgements

Questa pubblicazione è stata realizzata da ricercatrice con contratto di ricerca cofinanziato dall'Unione europea - PON Ricerca e Innovazione 2014-2020 ai sensi dell'art. 24, comma 3, lett. a, della Legge 30 dicembre 2010, n. 240 e s.m.i. e del D.M. 10 agosto 2021 n. 1062. Questa ricerca è stata anche finanziata dall'Università degli Studi di Parma attraverso l'azione Bando di Ateneo 2022 per la ricerca co-finanziata dal MUR-Ministero dell'Università e della Ricerca - D.M. 737/2021 - PNR - PNRR - NextGenerationEU.

9. References

Francisca Adoma Acheampong, Chen Wenyu, and Henry Nunoo-Mensah. 2020. Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports*, 2(7):e12189.

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 579–586, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Oscar Araque, Simona Frenda, Rachele Sprugnoli, Debora Nozza, and Viviana Patti. 2023. EMit at EVALITA 2023: Overview of the Categorical Emotion Detection in Italian Social Media Task. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, pages 37–44, Parma, Italy. AILC - Associazione Italiana di Linguistica Computazionale, Accademia University Press.

Federico Bianchi, Debora Nozza, and Dirk Hovy. 2021. FEEL-IT: Emotion and sentiment classification for the Italian language. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 76–83, Online. Association for Computational Linguistics.

Federico Bianchi, Debora Nozza, and Dirk Hovy. 2022. XLM-EMO: Multilingual emotion prediction in social media text. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 195–203, Dublin, Ireland. Association for Computational Linguistics.

Laura-Ana-Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2104–2119, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Sven Buechel and Udo Hahn. 2017. EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 578–585, Valencia, Spain. Association for Computational Linguistics.

Federico Cabitza, Andrea Campagner, and Valerio Basile. 2023. Toward a perspectivist turn in ground truthing for predictive computing. volume 37, pages 6860–6868.

Rafael A. Calvo and Sunghwan Mac Kim. 2013. Emotions in text: dimensional and categorical models. *Computational Intelligence*, 29(3):527–543.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, online. Association for Computational Linguistics.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 4040–4054, Online. Association for Computational Linguistics.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Michael Fischer. 1990. Perspectivism and literary theory today. *American Literary History*, 2(3):528–549.

Suzanne Keen. 2011. Introduction: Narrative and the Emotions. *Poetics Today*, 32(1):1–53.

Evgeny Kim and Roman Klinger. 2019. A survey on sentiment and emotion analysis for computational literary studies. Wolfenbüttel.

Vicki Liu, Carmen Banea, and Rada Mihalcea. 2017. Grounded emotions. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 477–483. IEEE.

Elena Maiolini et al. 2017. *Manzoni. Il linguaggio delle passioni*, volume 68. Franco Cesati.

Massimiliano Mancini. 2005. *Il romanzo dell'ironia. I Promessi Sposi nella critica*. Vecchiarelli.

Alessandro Manzoni. 2022. *The Betrothed: A Novel*. Modern Library. Translated by Michael F. Moore.

Saif Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184, Melbourne, Australia. Association for Computational Linguistics.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Emily Öhman, Marc Pàmies, Kaisla Kajava, and Jörg Tiedemann. 2020. XED: A multilingual dataset for sentiment analysis and emotion detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6542–6552, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Federico Pianzola, Simone Rebora, and Gerhard Lauer. 2020. Wattpad as a resource for literary studies. quantitative and qualitative examples of the importance of digital social reading and readers' comments in the margins. *PloS one*, 15(1):e0226708.

Flor Miriam Plaza-del Arco, Salud M Jiménez Zafra, Arturo Montejo Ráez, M. Dolores Molina González, Luis Alfonso Ureña López, and María Teresa Martín Valdivia. 2021. Overview of the EmoEvalEs task on emotion detection for Spanish at IberLEF 2021. Sociedad Española para el Procesamiento del Lenguaje Natural.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Ezio Raimondi. 1990. *La dissimulazione romanzesca: antropologia manzoniana*. Il Mulino.

Simone Rebora. 2023. Sentiment analysis in literary studies. a critical survey. *DHQ: Digital Humanities Quarterly*, 17(2).

James A Russell and Albert Mehrabian. 1974. Distinguishing anger and anxiety in terms of emotional response factors. *Journal of consulting and clinical psychology*, 42(1):79.

Jatinderkumar R Saini and Jasleen Kaur. 2020. Kāvi: An annotated corpus of punjabi poetry with emotion detection based on 'navrasa'. *Procedia Computer Science*, 167:1220–1229.

Klaus R Scherer. 1984. Emotion as a multicomponent process: A model and some cross-cultural data. Sage Publications, Inc.

Thomas Schmidt, Manuel Burghardt, and Katrin Dennerlein. 2018. Sentiment annotation of historic german plays: An empirical study on annotation behavior. In Sandra Kübler and Heike Zinsmeister, editors, *Proceedings of the Workshop on Annotation in Digital Humanities 2018 (annDH 2018)*, pages 47–52. Sofia, Bulgaria.

Thomas Schmidt, Katrin Dennerlein, and Christian Wolff. 2021. Towards a corpus of historical german plays with emotion annotations. In *3rd Conference on Language, Data and Knowledge (LDK 2021)*, pages 1–11. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Hendrik Schuff, Jeremy Barnes, Julian Mohme, Sebastian Padó, and Roman Klinger. 2017. Annotation, modelling and analysis of fine-grained emotions on a stance and sentiment detection corpus. In *Proceedings of the 8th Workshop on*

Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 13–23, Copenhagen, Denmark. Association for Computational Linguistics.

Rachele Sprugnoli. 2020. Multiemotions-it: A new dataset for opinion polarity and emotion analysis for italian. In *Proceedings of the Seventh Italian Conference on Computational Linguistics (CLiC-it 2020)*, pages 402–408. Accademia University Press.

Rachele Sprugnoli, Francesco Mambrini, Marco Passarotti, and Giovanni Moretti. 2023. The Sentiment of Latin Poetry. Annotation and Automatic Analysis of the Odes of Horace. *IJCoL. Italian Journal of Computational Linguistics*, 9(9-1):53–71.

Carlo Strapparava and Rada Mihalcea. 2008. Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1556–1560.

Marco Vassallo, Giuliano Gabrieli, Valerio Basile, Cristina Bosco, et al. 2020. Polarity imbalance in lexicon-based sentiment analysis. In *Proceedings of the Seventh Italian Conference on Computational Linguistics*, pages 1–7. CEUR.

Shibingfeng Zhang, Francesco Fernicola, Federico Garcea, Paolo Bonora, and Alberto Barrón-Cedeño. 2022. Ariemozione 2.0: Identifying emotions in opera verses and arias. *IJCoL. Italian Journal of Computational Linguistics*, 8(8-2):7–26.

Leveraging LLMs for Post-OCR Correction of Historical Newspapers

Alan Thomas[1, 2], **Robert Gaizauskas**[2], **Haiping Lu**[1, 2]
[1]Centre for Machine Intelligence, The University of Sheffield
[2]Department of Computer Science, The University of Sheffield
{alan.thomas, r.gaizauskas, h.lu}@sheffield.ac.uk

Abstract

Poor OCR quality continues to be a major obstacle for humanities scholars seeking to make use of digitised primary sources such as historical newspapers. Typical approaches to post-OCR correction employ sequence-to-sequence models for a neural machine translation task, mapping erroneous OCR texts to accurate reference texts. We shift our focus towards the adaptation of generative LLMs for a prompt-based approach. By instruction-tuning Llama 2 and comparing it to a fine-tuned BART on BLN600, a parallel corpus of 19th century British newspaper articles, we demonstrate the potential of a prompt-based approach in detecting and correcting OCR errors, even with limited training data. We achieve a significant enhancement in OCR quality with Llama 2 outperforming BART, achieving a 54.51% reduction in the character error rate against BART's 23.30%. This paves the way for future work leveraging generative LLMs to improve the accessibility and unlock the full potential of historical texts for humanities research.

Keywords: ocr, large language model, newspaper, historical text, digital humanities

1. Introduction

Historical newspapers are crucial primary sources for humanities research, providing valuable insights into past events, cultural perspectives and societal changes. Significant digitisation efforts have been undertaken to enhance accessibility to these sources by scanning newspaper pages and utilising Optical Character Recognition (OCR) technology to convert images into text. This content is then stored in searchable online databases with a prominent example being *British Library Newspapers* (Gale, 2024), a collection spanning 300 years of newspaper publishing in the United Kingdom.

Unfortunately, the OCR quality frequently suffers due to the distinct challenges presented by historical newspapers, such as degradation over time (bleed-through, ink spills, fading), inferior print quality, outdated typefaces and complex newspaper layouts (Holley, 2009). This significantly hampers the effectiveness of text mining techniques and keyword searches, hindering humanities scholars' ability to extract meaningful information. Addressing the issue of noisy OCR is crucial to unlocking the full potential of these primary sources. Post-OCR correction, which involves refining and enhancing the textual output generated by OCR technology, is a pivotal step in overcoming this challenge.

In recent years, the introduction of the Transformer model (Vaswani et al., 2017) has sparked a revolution in Natural Language Processing (NLP). Transformer-based architectures have consistently achieved state-of-the-art performance across a range of tasks, such as named entity recognition, sentiment analysis, question answering, and machine translation. Within this context, post-OCR correction has often been framed as a sequence-to-sequence neural machine translation problem (Nguyen et al., 2021), with Transformer-based models trained to map erroneous OCR text to the accurate reference text.

The emergence of foundation models marks another significant milestone in NLP research. Generative large language models (LLMs), exemplified by GPT-3 (Brown et al., 2020), are trained on massive datasets and contain billions of parameters. This enables them to produce coherent and contextually relevant responses to a given prompt, showcasing remarkable language understanding capabilities and adaptability for downstream tasks across different domains. Given these factors, we believe it is worth exploring the potential of such models to perform post-OCR correction.

In this work, we focus on the post-OCR correction of BLN600, an open-source dataset of 19th century newspaper articles, written in English (Booth et al., 2024). This dataset contains OCR text sourced from *British Library Newspapers* along with manually re-keyed human transcriptions. We benchmark and compare two different approaches to post-OCR correction. Firstly, we adopt the prevalent approach in literature and fine-tune BART (Lewis et al., 2020), a sequence-to-sequence model, for a neural machine translation task. Secondly, we explore the potential of instruction-tuning Llama 2 (Touvron et al., 2023), an open-access foundation model, for a prompt-based approach. Through this comparison, we aim to demonstrate the capabilities of the latter approach for improving the OCR quality of digitised historical newspapers. Llama 2 outperforms BART, reducing the character error rate of our test set by 54.51% compared to 23.30%.

LT4HALA 2024@LREC-COLING-2024, pages 116–121

2. Related Work

Since the development of OCR technology, post-OCR correction has been a critical challenge. As outlined by Nguyen et al. (2021), post-OCR correction approaches can broadly be categorised into three main types: manual, isolated-word, and context-dependent. Manual approaches involve direct human intervention to correct errors in OCR generated text, achieving high accuracy at the cost of significant time and labour. Isolated-word approaches focus on examining each word separately through strategies such as merging outputs from different systems, modelling frequent errors made by OCR engines or dictionary-based correction. Context-dependent approaches consider the text around the error, typically outperforming isolated-word approaches with language models, feature-based methods and sequence-to-sequence models falling into this category.

Post-OCR correction of historical documents has seen recent coverage in literature after the International Conference on Document Analysis and Recognition (ICDAR) held two competitions on post-OCR correction (Chiron et al., 2017; Rigaud et al., 2019), involving error detection and error correction tasks. The competitions introduced parallel corpora, with the ICDAR2017 corpus comprising 12M characters from English and French texts and the ICDAR2019 corpus expanding to 22M characters across multiple European languages. A key feature of the datasets is that the OCR text is aligned at character level with the ground truth using special symbols ("@" for padding, "#" for ignoring) to ensure they are of the same length.

The first competition was dominated by statistical and neural machine translation methods (Chiron et al., 2017), with Char-SMT/NMT emerging as the winner with an ensemble of character-level translation models (Amrhein and Clematide, 2018). In the second competition, Clova AI's Context-based Character Correction method achieved the best performance (Rigaud et al., 2019), making use of a pre-trained multilingual BERT. Since the conclusion of the competitions, Ramirez-Orta et al. (2022) attained a new state-of-the-art performance on the ICDAR2019 corpus by combining corrections of character-level sequence-to-sequence models using a voting scheme. Soper et al. (2021) fine-tuned BART for sentence-level correction, achieving a comparable performance on the ICDAR2017 corpus with a simpler, single-step approach.

The works above indicate the prevalent approach to post-OCR correction is sequence-to-sequence neural machine translation, with pre-trained models being leveraged more recently. To our knowledge, we are the first to explore how generative LLMs can be prompted for post-OCR correction.

3. Methodology

In this section, we outline our methodology, providing background on the BLN600 dataset, as well as details of BART, Llama 2, and their respective training processes. We had planned to assess the effectiveness of our approach on the ICDAR corpora for post-OCR correction. However, these datasets contain excerpts from literary works that are available online and may be present in the training data of Llama 2, leading to potential data contamination and evaluation issues (Sainz et al., 2023).

3.1. BLN600

BLN600 is a parallel corpus of 19th century newspaper machine/human transcription (Booth et al., 2024). The dataset contains OCR excerpts from *British Library Newspapers Parts I-II (1800-1900)* (Gale, 2024), along with high-quality manually rekeyed human transcriptions from the source images. Comprising 600 samples, the articles are sourced from six different publications, published between the decades spanning the 1830s and 1890s, encapsulating a significant period of societal and cultural transformation.

Due to the acquisition process, BLN600 largely focuses on crime-related news from London publications, detailing criminal cases, court proceedings and punishments. The dataset notably reflects 19th century vocabulary and linguistic conventions, with abbreviations like "ult." (ultimo) and "inst." (instant), as well as old currency terms such as "£ s. d." (pounds, shillings, and pence). Additionally, changes in spelling conventions over time add another layer of complexity. In total, both the OCR text and ground truth contain around 300K tokens and 1.7M characters each.

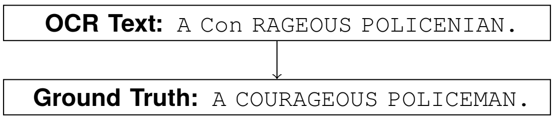

OCR Text: A Con RAGEOUS POLICENIAN.
Ground Truth: A COURAGEOUS POLICEMAN.

Figure 1: Example of input/output sequences

Unlike ICDAR, the OCR text and ground truth are not aligned at character level in BLN600 and can vary significantly in length, which affects how the data can be prepared as input to the model. We prepare a dataset of sequence pairs by splitting the ground truth into segments. These segments are usually individual sentences but can also be shorter article titles or longer passages like quotes. This approach allows our models to accommodate sequences of varying lengths. For each ground truth segment, the corresponding OCR text is then gathered using a search algorithm to create a dataset of source and target texts, as illustrated in Fig. 1 where the sequence is an article title.

After creating sequence pairs, we prepare training and evaluation sets, ensuring sequences from the same sample are kept in different sets. Table 1 provides a breakdown of the sets along with details of the mean μ and standard deviation σ in character error rate. Character error rate (CER) is used to evaluate the performance of text recognition systems such as OCR engines by computing the Levenshtein distance between the recognised text and the reference text and dividing it by the total number of characters in the reference text to provide a measure of accuracy. Levenshtein distance counts the number of edits required to transform one string into another with substitutions (replacing one character with another), insertions (adding a new character) and deletions (removing an existing character). We include 1968 perfectly correct sequence pairs with a CER of 0 (15% of the entire dataset) across our sets, such that our models learn to recognise and preserve accurate OCR outputs.

	# sample	# sequence	μ CER	σ CER
Total	600	13,192	0.0771	0.1216
Train	480	10,400	0.0753	0.1175
Test	120	2,792	0.0840	0.1354

Table 1: BLN600 breakdown with CER statistics

3.2. BART

BART (Bidirectional and Auto-Regressive Transformers) is a language model that is pre-trained on multiple denoising tasks, enabling it to reconstruct text from corrupted inputs (Lewis et al., 2020). This is achieved through pre-training tasks including token masking, token deletion, text infilling, sentence permutation and document rotation. BART uses a standard sequence-to-sequence architecture, combining BERT's bidirectional encoder (Devlin et al., 2019) for language understanding and GPT's autoregressive decoder (Radford et al., 2019) for generative tasks, making it particularly suited to summarisation and translation tasks.

As Soper et al. (2021) highlight, BART's pre-training makes it well suited for post-OCR correction given the similarities between its denoising tasks and the correction of OCR errors. Additionally, the input to the encoder does not need to be aligned with the decoder output at character level, enabling it to deal with the unaligned OCR text and ground truth sequences in BLN600.

We train BART for a neural machine translation task, operating on sequence pairs as illustrated in Figure 1, where the OCR text is the input and the ground truth is the target. We make use of Hugging Face Transformers library (Wolf et al., 2020), fine-tuning both the 'base' (140M parameters) and 'large' (400M parameters) versions.

3.3. Llama 2

Llama 2 is a family of pre-trained and fine-tuned LLMs released by Meta AI (Touvron et al., 2023). It is a decoder-only, generative LLM with a context length of 4096, pre-trained on a mix of publicly available sources, comprising 2 trillion tokens. We opted to use Llama 2 due to its open-access nature and availability of various versions. The models come in three different parameter sizes (7B, 13B, 70B). The pre-trained model ('base') is a causal language model, designed to predict the next word in a sequence, which can be adapted for various natural language generation tasks. The fine-tuned model ('chat') is designed for assistant-like chat and optimised for dialogue applications through reinforcement learning from human feedback.

Using the sequence pairs in our train set, we create a new instruction-tuning dataset, following the Alpaca format with instruction, input and response fields (Taori et al., 2023), using a clear and simple prompt for correcting OCR errors, as illustrated in Fig. 2. We use Hugging Face Transformers to train LoRA (Hu et al., 2022) adaptors for both the 7B and 13B 'base' versions of Llama 2 on this instruction-tuning dataset, reducing the number of trainable parameters to achieve efficient fine-tuning.

```
### Instruction:
Fix the OCR errors in the provided text.

### Input:
A Con RAGEOUS POLICENIAN.

### Response:
A COURAGEOUS POLICEMAN.
```

Figure 2: Instruction-tuning dataset sample

4. Results

After training, we generate predictions over the test set. The OCR text is directly passed as input for BART. For Llama 2, we use the prompt shown in Fig. 2, excluding the response on the last line.

Table 2 illustrates some of the examples of corrections made by our top performing model, Llama 2 13B, across different error types. Substitutions and deletions are the most common type of error and easiest for our models to correct. Insertions are rarer, only occurring in cases of extreme fading in the source image, and more difficult to correct due to the dependence on the surrounding context. Correction of misnamed entities including names, locations and organisations is of particular interest to improve text mining results. The performance on these errors varies more, depending on the level of corruption. When OCR systems hallucinate and produce text that is unintelligible even to humans, it is understandable that our models struggle.

Error Type	OCR Text	Ground Truth	Model Correction
Substitution	CRUELTY TO A RACE H**ons**E.	CRUELTY TO A RACE H**ORS**E.	CRUELTY TO A RACE H**ORS**E.
Insertion	It vas impossible for a magitrate to adjtdiete in such	It was impossible for a magistrate to adjudicate in such **a matter.**	It was impossible for a magistrate to adjudicate in such **a case.**
Deletion	She spoke**- T 5** once only, and that was while the doctor's evidence **p i** was given.	She spoke once only, and that was while the doctor's evidence was given.	She spoke once only, and that was while the doctor's evidence was given.
Misnamed Entity	_ **Mat. Teix TROuTr31accK**, the coroner for Westminster,	**MR. JOHN TROUTBECK**, the coroner for Westminster,	**MATTHEW RUTHVEN**, the coroner for Westminster,
Hallucination	A BAD , .0- ? aff-a. . t , -	A SAD CASE.	A BAD AFFRAY.

Table 2: Llama 2 13B corrections on different error types

We measure the performance of our models for post-OCR correction by computing the percentage reduction in CER, shown in Formula (1), where s is the source OCR text, t is the target ground truth and p is the model's prediction. As shown in Table 3, we calculate the average percentage reduction in CER across our test set for BART and Llama 2. To account for variance, each value represents the mean across five training and evaluation iterations.

$$CER_{reduction} = \left(\frac{CER(s,t) - CER(p,t)}{CER(s,t)} \right) \times 100 \tag{1}$$

Model	Size	CER Reduction (%)
BART	140M	14.60
	400M	23.30
Llama 2	7B	43.26
	13B	54.51

Table 3: Comparison of model performance

BART achieves respectable results with its 'large' variant attaining a notable 23.30% reduction. Llama 2 significantly outperforms BART, particularly the 13B model, which is over twice as effective with a score of 54.51%. However, foundation models like Llama 2 are predominantly trained on English data and adapting such models for post-OCR correction in other languages presents an additional challenge. In contrast, multilingual sequence-to-sequence models are widely available for this purpose including mBART (Liu et al., 2020).

Leveraging a generative LLM like Llama 2 also presents a notable advantage in its ability to adapt well to downstream tasks with a limited amount of instruction-tuning data (Zhou et al., 2023). On the contrary, machine translation models, including those that leverage pre-trained models like BART, are known to rely on large volumes of parallel data (Xu et al., 2024). We explore this phenomenon by dividing our original train set shown in Table 1 into six subsets of 80 samples. We evaluate the performance of BART and Llama 2 six times, incorporating sequences from an additional subset each time to increase the amount of training data. As shown in Fig. 3, BART improves significantly with more training data whilst Llama 2 exhibits strong performance from the outset. When working with limited training data, foundation models offer a major advantage given their extensive pre-training.

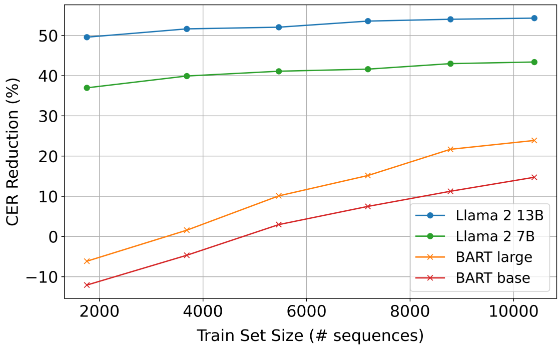


Figure 3: Performance versus train set size

5. Conclusion

In this work, we performed post-OCR correction of BLN600, a dataset of 19th century British newspaper articles. We compared the performance of a neural machine translation method to a prompt-based approach leveraging a generative LLM. We showcase Llama 2's ability to detect and correct OCR errors, significantly outperforming BART.

Moving forward, we believe that post-OCR correction for digitisation projects should leverage foundation models fine-tuned on small, curated datasets of genre-adjacent and period-specific text. In future work, we intend to build an assistant model capable of explaining error corrections with the 'chat' version of Llama 2. This would enhance the model's reliability and trustworthiness whilst enabling human verification and intervention for difficult errors. We plan to explore the possibility of quantifying the model's confidence in its corrections, which could then be used to flag a correction for review.

6. Availability Statements

BLN600 is publicly accessible at https://doi.org/10.15131/shef.data.25439023. The code is available on GitHub at https://github.com/alanbijuthomas. The fine-tuned models will be released on Hugging Face at https://huggingface.co/pykale.

7. Acknowledgments

This work was supported by the Centre for Machine Intelligence and the Digital Humanities Institute at the University of Sheffield. The authors would like to thank Callum Booth for his research into the OCR quality of *British Library Newspapers* and preparation of BLN600, which have enabled this work.

8. References

Chantal Amrhein and Simon Clematide. 2018. Supervised OCR Error Detection and Correction Using Statistical and Neural Machine Translation Methods. *Journal for Language Technology and Computational Linguistics*, 33(1):49–76.

Callum Booth, Alan Thomas, and Robert Gaizauskas. 2024. BLN600: A Parallel Corpus of Machine/Human Transcribed Nineteenth Century Newspaper Texts. In *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Guillaume Chiron, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2017. ICDAR2017 Competition on Post-OCR Text Correction. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 1423–1428.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Gale. 2024. British Library Newspapers. https://www.gale.com/intl/primary-sources/british-library-newspapers.

Rose Holley. 2009. How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. *D-Lib Magazine*, 15(3/4).

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Thi Tuyet Hai Nguyen, Adam Jatowt, Mickael Coustaty, and Antoine Doucet. 2021. Survey of Post-OCR Processing Approaches. *ACM Computing Surveys*, 54(6):1–37.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI.

Juan Antonio Ramirez-Orta, Eduardo Xamena, Ana Maguitman, Evangelos Milios, and Axel J. Soto. 2022. Post-OCR Document Correction with Large Ensembles of Character Sequence-to-Sequence Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11192–11199.

Christophe Rigaud, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2019. ICDAR 2019 Competition on Post-OCR Text Correction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1588–1593.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787. Association for Computational Linguistics.

Elizabeth Soper, Stanley Fujimoto, and Yen-Yun Yu. 2021. BART for Post-Correction of OCR Newspaper Text. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 284–290. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpaca: A Strong, Replicable Instruction-Following Model. Technical report, Centre for Research on Foundation Models, Stanford University.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. Technical report, Meta AI.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Kaiser Łukasz, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, volume 30, page 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A Paradigm Shift in Machine Translation: Boosting Translation Performance of Large Language Models. In *International Conference on Learning Representations*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. LIMA: Less Is More for Alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021.

LLM-based Machine Translation and Summarization for Latin

Martin Volk, Dominic P. Fischer, Lukas Fischer, Patricia Scheurer, Phillip B. Ströbel
Department of Computational Linguistics, University of Zurich
volk@cl.uzh.ch

Abstract

This paper presents an evaluation of machine translation for Latin. We tested multilingual Large Language Models, in particular GPT-4, on letters from the 16th century that are in Latin and Early New High German. Our experiments include translation and cross-language summarization for the two historical languages into modern English and German. We show that LLM-based translation for Latin is clearly superior to previous approaches. We also show that LLM-based paraphrasing of Latin paragraphs from the historical letters produces English and German summaries that are close to human summaries published in the edition.

Keywords: Large Language Models, Machine Translation, Latin, Early New High German, GPT

1. Introduction

The advent and wide accessibility of large language models (LLMs) with their inherent multilingual abilities has founded a new paradigm for machine translation (MT). LLM-based MT is similar to neural MT but has advantages for low-resource languages because of cross-language knowledge transfer and the possibility of targeted translation suggestions. In this paper we explore GPT-4 (OpenAI, 2023) as MT system for Latin to English and to German. We tested GPT-4's MT performance on letters from the 16th century that are in Latin and Early New High German (ENH-German).

The MT community site[1] documents that MT for Latin is "supported by 10 APIs". We checked the corresponding websites and found that five of these allow for online testing: Google Translate, LingvaNex, ModernMT, Niutrans and Yandex, for all of which Latin is one among more than 100 supported languages. Our tests show that translation quality for Latin to English and German is low for most of these systems. For a first glimpse of the results see table 1. We will detail the figures in section 4.1.

Fischer et al. (2022) described a neural MT system for Latin to German translation that outperformed Google Translate on their test set. In the meantime the situation has changed. Recent multilingual LLMs show surprising performance for machine translation.

This paper proves that GPT-4 produces superior MT quality for Latin to German and Latin to English if prompted appropriately. We also show that the same technology is able to produce paraphrases of the historical letters which compare favorably with human-written summaries.

2. Previous Work on LLMs for Latin

Work on using language models for Latin started with Bamman and Burns (2020) who built Latin-BERT on more than 600 million words. This established a new state of the art for part-of-speech tagging for Latin and for predicting missing text. Following up, Nehrdich and Hellwig (2022) used the Latin BERT embeddings for PoS tagging and dependency parsing for Latin. Lendvai and Wick (2022) used Latin BERT for Word Sense Disambiguation. They confirm that the contextualized BERT representations finetuned on *the Thesaurus Linguae Latinae*[2] score better than static embeddings from a bidirectional LSTM classifier.

With the advent of ChatGPT the question arose: How good is the GPT technology for historical languages? And why is it so good? Burns (2023) addresses these questions in his blog post and estimates that GPT-3 has been trained on more than 300 million tokens of Latin text. This is only a small fraction of its total training corpus but enough to model the language for high-performance on tasks like part-of-speech tagging, spelling and grammar correction for Latin texts.

Riemenschneider and Frank (2023) investigated the use of LLMs for Latin and Ancient Greek. They focus on Greek, but also built a multilingual model with English and Latin (with roughly 200 million tokens in each language as training corpora). For Latin they evaluated their model against the EvaLatin 2022 dataset (Sprugnoli et al., 2022) and report superior performance for part-of-speech tagging and lemmatization.

LLMs are trained on large amounts of text, most of which is typically in English. But even small amounts of other languages in the training data enable the system to respond in multiple languages and to learn to translate. Briakou et al. (2023) find

[1] https://machinetranslate.org

[2] https://thesaurus.badw.de/

that only 1.4% of training instances for the PaLM are bilingual which still results in good translation performance for medium-resource languages like Bulgarian, Hebrew, and Greek (Latin not included in this study), especially for MT into English. Fine-tuning LLMs on translation tasks results in improved MT performance, as Xu et al. (2023) showed for LLaMA-2.

GPT-based MT has been evaluated by various researchers. Laskar et al. (2023) report that ChatGPT scores slightly worse than the state-of-the-art for MT between high resource languages like English and French, but it is better than previous systems in translating Romanian into English and French into German.

We are the first to present a systematic evaluation of LLM-based MT and summarization for Latin and Early New High German.

3. The Corpus of Letters in Latin and ENH-German

We work with a large corpus of 16th-century letters (Volk et al., 2022; Ströbel et al., 2024). 3100 have been professionally edited and another 5400 have been manually transcribed. The letters include historical characters (like ę, ů, aͤ, oͤ). Abbreviations have been spelled out by the transcribers (e.g. the greeting *S et p in domino Iesu* has been completed into *S[alutem] et p[acem] in domino Iesu*, EN: *Greetings and Peace in the Lord Jesus*). Paragraph boundaries are set by the transcribers, sentence boundaries have been automatically added. Three quarters of the letters are in Latin, the rest in ENH-German, many letters contain code-switching between the two languages. The letters contain occasional sentences in Greek. All sentences have been automatically assigned a language tag based on a self-trained language identifier that is able to distinguish between ENH-German and Latin with high accuracy (see (Volk et al., 2022)).

The letters are part of the correspondence to and from the Zurich reformer Heinrich Bullinger. They deal with politics, theological debates, regional and European news as well as education and family matters. The letters thus give a first-hand view into the life 500 years ago. The correspondence network extended from Zurich throughout Germany towards Denmark, England, and Poland. Some letters traveled more than 1000 km.

4. Experiments with LLM-based MT

4.1. Evaluation against a Test Set

We used the test set of Fischer et al. (2022) which consists of 8 letters which have been manually translated into German by a domain expert. This test set focuses on Latin letters, but contains one sentence that is code-switching from Latin into ENH-German *Indixit dry musterpletz: Fuͤssen, Werdt und Nördlingen* (EN: *He designated three recruiting places: Füssen, Donauwörth and Nördlingen*).

These 8 letters sum up to a total of 121 Latin sentences, some of which are short greetings, others are as long as 47 words. The whole test set consists of 1240 words on the Latin side and 1768 words in the corresponding human-translated sentences in German.

In order to be able to re-use the test set for MT into English we automatically translated the human-translated German sentences into English with GPT-4.

We then translated the test set with Google Translate and the other online MT systems from Latin into German and into English in order to obtain the baseline scores. In a second step we fed the complete test set to GPT-4 with a single prompt: "Translate the following text from Latin into L" where L was first German and then English.

The resulting scores are in table 1: In translating Latin to German, GPT-4 outperforms Google Translate by close to 10 BLEU points on the test set. The other online MT systems score clearly worse than Google Translate both when measured with BLEU and with ChrF.[3]

Fischer et al. (2022) had reported a BLEU score of 19.5 for their own system and 17.07 for Google Translate. When testing Google Translate now, we obtain a score of 17.53, which is marginally higher. This means that Google Translate has not improved much for Latin MT in recent years. However, GPT-4 surpasses these results significantly, reaching a BLEU score of 27.07 for Latin to German MT on the test set (see table 1 for an overview).

We observe a similar quality increase in translating from Latin to English. Google Translate reached a BLEU score of 25.22 for this language direction, while GPT-4 again betters it considerably, reaching 34.50. This is an enormous improvement. Table 2 shows the differences in translation quality for an example sentence from our test set.

The discrepancy between English and German can be attributed to two major reasons:

1. English is by far the highest resource language on the internet, and many researchers reported better MT into English than into other languages (cf. section 2 above).

2. We translated the German reference translation into English using GPT-4, which may in-

[3] BLEU (Papineni et al., 2002) is a precision-oriented word n-gram overlap metric which is often used in MT evaluation. ChrF (Popović, 2015) is a character n-gram metric which uses precision and recall.

MT System	Languages	MT Latin into German		MT Latin into English	
		BLEU	ChrF	BLEU	ChrF
GPT-4	unknown	**27.07**	**50.55**	**34.50**	**54.6**
Google Translate	134	17.53	43.23	25.22	47.48
LingvaNex	109	12.08	37.54	17.72	39.76
Yandex Translate	102	11.36	35.35	12.64	35.43
ModernMT	200	9.78	32.42	13.56	34.2
Niutrans	449	4.45	26.8	5.52	26.9

Table 1: BLEU and ChrF scores when translating the Latin test set (121 sentences) into German and English. The first column has the number of supported languages per system.

Original Latin	**Quid sibi hęc societas velit, facile divinari potero.**
Human Reference German	Was dieses Bündnis bedeutet, kann ich mir leicht vorstellen.
Human Reference English (transl. from DE by GPT-4)	What this alliance means, I can easily imagine.
MT System	**Translation**
GPT-4	What this alliance means, I can easily guess.
GoogleTranslate	What this company wants for itself, I can easily guess.
LingvaNex	What society wants for itself here is that I will be able to be divined easily.
Niutrans	I'm afraid it's hard to predict why Szczesny himself chose to participate in the league.

Table 2: A Latin sentence taken from a letter of Johannes Gast to Heinrich Bullinger, 1. April 1544 (see https://www.bullinger-digital.ch/letter/11930), translated by different translation systems, ordered by their automatic evaluation scores; with GPT-4 performing best and Niutrans worst.

troduce a bias, as the English translation may now be skewed towards a GPT-4 style of writing. When used as a reference for the evaluation of the Latin-English translations, that bias might lead to higher BLEU scores for GPT-4. As the BLEU score increase between Google Translate and GPT-4 remains approximately the same for both language pairs Latin-German and Latin-English, we conclude that this bias cannot be the decisive factor.

4.2. Evaluation against Paragraph Summaries

Reference translations are tedious and costly to create. With GPT having proven its quality in translation from Latin to both German and English, we investigated whether we can use letter summaries to evaluate GPT-4 translations.

For each of the 3100 edited letters we have a summary in German which was written by experts of the Institute for Swiss Reformation Studies. For the initial volumes of the edition, which date back to the 1970s, the summaries consisted of a few sentences or paragraphs. Over time the summaries increased in length. The three most recent volumes of the letter edition (published in the years 2017 to 2022, cf. Gäbler et al. (1973–2022)) contain paragraph-by-paragraph summaries that can be seen as shortened paraphrases. The alignment between the Latin paragraph in the letter text and the German summary is given. For an example letter with summaries see appendix A.

We used 10 medium-sized letters (5-7 paragraphs each) in Latin from the volume 18 of the edition, where the human-written summaries are paragraph-by-paragraph. Since the human summaries in this volume are close to the letter text we hypothesized that the summaries could be used as reference translations.

With this setup GPT-4 achieved a low 4.93 BLEU points when we compare the automatic translation to the human summary in German. In analogy to our test set evaluation we also translated the human summaries from German into English with GPT-4. GPT-4 MT from Latin to English then results in 6.80 BLEU. Google Translate resulted in 3.43 BLEU for German and 5.84 for English. Interestingly, the MT scores are slightly higher when we translate the summaries from German into English with DeepL, which proves that GPT-4 translation DE-EN of the reference texts does not favor the MT results LA-EN towards GPT-4. See table 3 for the results.

Evaluating GPT-4 MT against the human summaries shows again that GPT-4 clearly outperforms Google Translate. But the scores differ by few BLEU points only and do not show the GPT-4 advantage as clear as with the test set.

5. LLM-based Summarization

In the previous section we tested whether the German summaries in the letter edition may serve as reference translations. Here we extend this idea to check whether GPT-4 can produce English or

Model	MT LA into DE	MT LA into EN (GPT-4)	MT LA into EN (DeepL)
Google Translate	3.43	5.84	6.59
GPT-4	**4.93**	**6.80**	**7.47**

Table 3: BLEU scores when translating 10 Latin letters into German and English, evaluated against the human summary in German, and a machine-translated summary (DE-EN) in English

German summaries for Latin and ENH-German letters.

In this experiment, we used the same 10 Latin letters as above, as well as 10 ENH-German letters from the edition. We prompted GPT-4 to produce a paragraph-by-paragraph summary of the given letter in the following way: "I have this letter by {sender} to {addressee} with {nr} paragraphs: {original_letter}. For each paragraph, write a summary in English from a third-person perspective."

We evaluated again by comparing the GPT-4 output with the human summary in German and the machine-translated summary (DE-EN) in English. When summarizing in German, GPT-4 achieves a BLEU score of 6.23 for the Latin letters and 5.45 for the letters in ENH-German.

In order to evaluate the summarization into English, we used both GPT-4 and DeepL to translate the human summaries from modern German into English and used these translations as reference. For Latin-English, GPT-4 now scores 9.98 on the DeepL reference translation and 10.40 on the GPT-4 reference translation. For ENH-German to English, the scores are 7.75 on the DeepL translation and 8.48 on the GPT-4 translation.

The BLEU scores for the automatic summaries are low, but confirm that GPT's output in English is of slightly higher quality than in German. A comparison of the scores for the ENH-German letters with the Latin letters is not possible. These are different letters.

Even though the summarization scores are low, the summaries look very good. In order to check the quality and assess their usefulness, we conducted a manual evaluation of GPT's automatically produced German summaries, using the following criteria. We checked for each paragraph whether

- the names (persons, locations) that are mentioned in the human summary are also contained in the generated GPT summary
- the events and times of the human summary are included in the generated GPT summary
- the information from the human summary is contained **completely** in the generated GPT summary
- the information of the human summary is **correctly** contained in the generated GPT summary

Three annotators compared and judged the human-written summaries to the GPT-produced German summaries paragraph-by-paragraph.

This evaluation yielded the results in table 5. Names are well represented in the generated GPT summaries, in particular person names. GPT-4 shows some issues with consistency: "Schweiz" (EN: *Switzerland*) is repeatedly used synonymously to "Eidgenossenschaft" (EN: *confederation*), which historically does not make sense. The average human evaluation score with regards to the names is 47.2 out of 58.

Times and Events were best captured by GPT with a score of 54.5 out of 58. Dates and temporal expression were accurately transferred into the summary. With regards to completeness, human evaluation yields a score of 48.2 out of 58, showing slight differences between the generated and the reference summary. It is to be noted, however, that completeness is sometimes subjective, since the expert editors weigh events by importance and thus decide whether or not to include them in the summaries. In a few cases, GPT-4 provided additional information that was pertinent, yet not contained in the human summary. Correctness was the lowest of the 4 metrics, with 43.5 out of 58 points. In some cases, potentially sensitive or offensive information was not correctly rendered, possibly due to censoring by GPT. Moreover, mistranslation of a few words or phrases led to opposite interpretation (e.g. "mirari" as "admire" instead of "be astounded").

We also noted positively that GPT-4's summaries of our test letters are free of any hallucinations: all information that is found in GPT's summaries is derived from the original Latin letter.

6. Advantages of LLM-based MT

Our results show clear advantages of LLM-based MT quality for Latin and ENH-German over the previous generation of neural MT systems. In addition, there are some technical aspects that speak in favor of LLM-based MT.

6.1. Steering the Translation

One striking advantage of LLM-based MT is the possibility for the user to suggest the translation of specific terminology to the LLM. For example, we observed that GPT-4 translates the Latin word *caesar* with the same word in English. However, in our

Model	ENH-German	Latin	ENH-German	Latin
	into German		into English (GPT-4 / DeepL)	
GPT-4	5.45	6.23	8.48 / 7.75	10.40 / 9.98

Table 4: BLEU scores when summarising 10 ENH-German and 10 Latin letters paragraph-wise into German and into English

	Names	Times & Events	Complete	Correct
Judge 1	44	56	49	43.5
Judge 2	45	52	46.5	36.5
Judge 3	52.5	55.5	49	50.5
Average	47.2	54.5	48.2	43.5

Table 5: Evaluator scores for the four evaluation categories on automatic summarization. The maximum points per category is 58, which means 1 point each for the 58 paragraphs in the test letters.

context *caesar* refers to the German emperor (Karl V. until 1556, and Ferdinand I. afterwards). Adding the instruction “Translate ‘caesar’ with ‘emperor’ ” to the GPT-4 prompt is enough to steer the translation of *caesar* with its inflected forms *caesarem, caesare* etc. to be translated in the desired way. If needed, the translation instruction can be enriched with world knowledge, e.g. by specifying the name of the respective emperor.

We observed such rare mistranslations not only with nouns but also with names. GPT-4 knows a surprising number of Latin city names and translates them correctly into modern day equivalents (e.g. Basilea → Basel, Lutetia → Paris, Tigurinę → Zurich). Still it gets confused when old names are homographs to modern names. In our case of 16th century Latin *Argentina* refers to the city of *Strasbourg* but is often mis-translated as the country name. The simple instruction “Translate ‘Argentina’ with ‘Strasbourg’ ” solves this problem for us, since the country name does not occur in our texts.

In the experiments reported in this paper we did not use the option of steering the translation.

6.2. Preserving XML Tags

Our corpus is annotated in XML for sentence boundaries, person and place names, for footnotes and page breaks. In order to use this valuable information after translation, the XML tags need to preserved in the target language. On a side project we experimented with MT for Latin to English with XML tags for sentence boundaries and names. We find that they are well-preserved when we translate with GPT-4. This requires specific prompting to inform the system about the XML in the input and the request for preserving the tags in the output.

7. Conclusion

GPT-4’s performance on Machine Translation for historical languages is impressive. We experimented with letters from the 16th century that are partly in Latin and partly in Early New High German. The quality for translating both languages to modern English and German is high, much higher than with previous neural MT technology. We measure an improved score of plus 10 BLEU points for both Latin to English and Latin to German LLM-based MT over Google Translate on a test set of 121 sentences. This is a huge improvement of the state-of-the-art.

In a second round of experiments we evaluated LLM-based paragraph-wise summarization against expert-written summaries. Our manual evaluation showed that the automatically generated summaries capture names, events and other pieces of information accurately.

We deal with letters that contain a lot of code-switching between Latin and ENH-German. Unlike previous MT the new generation of LLM-based MT is robust against language mix, which is a big advantage. We will investigate this aspect in more detail in future work.

This paper focused on GPT-4 as a prominent LLM example. Future work should compare GPT’s performance to other LLMs like LLama or Google Gemini. There, it will also be interesting to check how translation quality can be improved further by fine-tuning the LLMs to the Latin and ENH-German translation task.

We argued that it is easy to feed special terminology to the system in order to influence the translation (see (Bogoychev and Chen, 2023) for a systematic study). We plan to investigate the steering of the translation as a finetuning step by automatically identifying terms that require special translation instructions.

8. Acknowledgements

We gratefully acknowledge project funding provided by various sponsors through the UZH Foundation (see `www.bullinger-digital.ch/about`).

This paper drew inspiration from COST Action Multi3Generation (CA18231) supported by COST (European Cooperation in Science and Technology).

9. Bibliographical References

David Bamman and Patrick J. Burns. 2020. Latin BERT: A contextual language model for classical philology. In *ArXiv*.

Nikolay Bogoychev and Pinzhen Chen. 2023. Terminology-aware translation with constrained decoding and large language model prompting. In *Proceedings of the Eighth Conference on Machine Translation (WMT)*, pages 890–896, Singapore. Association for Computational Linguistics.

Eleftheria Briakou, Colin Cherry, and George Foster. 2023. Searching for needles in a haystack: On the role of incidental bilingualism in PaLM's translation capability. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 9432–9452, Toronto, Canada.

Patrick J. Burns. 2023. Research Recap: How much Latin does ChatGPT 'know'? Online blog post of May 19th, 2023.

Lukas Fischer, Patricia Scheurer, Raphael Schwitter, and Martin Volk. 2022. Machine translation of 16th century letters from Latin to German. In *Proceedings of 2nd Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) at LREC-2022*, pages 43–50, Marseille.

Ulrich Gäbler, Endre Zsindley, Kurt Maeder, Matthias Senn, Kurt Jakob Rüetschi, Hans Ulrich Bächtold, Rainer Heinrich, Alexandra Kess, Christian Moser, Reinhard Bodenmann, Judith Steiniger, and Yvonne Häfner, editors. 1973–2022. *Heinrich Bullinger Briefwechsel*. Heinrich Bullinger Werke. Theologischer Verlag Zürich.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 431–469.

Piroska Lendvai and Claudia Wick. 2022. Finetuning Latin BERT for Word Sense Disambiguation on the Thesaurus Linguae Latinae. In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, pages 37–41.

Sebastian Nehrdich and Oliver Hellwig. 2022. Accurate dependency parsing and tagging of Latin. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2022) at LREC*, pages 20–25, Marseille.

OpenAI. 2023. GPT-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, Philadelphia, PA, USA.

Maja Popović. 2015. chrf: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Frederick Riemenschneider and Anette Frank. 2023. Exploring large language models for classical philology. In *Proceedings of the 61st Annual Meeting of the ACL*, pages 15181–15199.

Rachele Sprugnoli, Marco Passarotti, Flavio Massimiliano Cecchini, Margherita Fantoli, and Giovanni Moretti. 2022. Overview of the EvaLatin 2022 evaluation campaign. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages (LREC)*, pages 183–188.

Phillip Benjamin Ströbel, Lukas Fischer, Raphael Müller, Patricia Scheurer, Bernard Schroffenegger, and Martin Volk. 2024. Multilingual workflows in Bullinger Digital: Data curation for Latin and Early New High German. *Journal of Open Humanities Data*, 10.

Martin Volk, Lukas Fischer, Patricia Scheurer, Raphael Schwitter, Phillip Ströbel, and Benjamin Suter. 2022. Nunc profana tractemus. Detecting code-switching in a large corpus of 16th century letters. In *Proceedings of LREC-2022*, pages 2901–2908, Marseille.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models. In *ArXiv*.

A. Example of a Mixed Language Letter

Letter in ENH-German and Latin	Human Summary (German) from the Heinrich Bullinger edition	GPT-4 Summary (English)
S. Gratulor tibi, honorande mi Myconi, ob recuperatam sanitatem tuam, quam dominus velit esse diuturnam, ut diu utilis esse pergas ecclesiae suae; quo etiam omnia tua studia convertas!	[1] Gut, dass es Myconius wieder besser geht! Der Herr möge ihn noch lange seiner Kirche erhalten. Ihr soll er sich völlig widmen!	Heinrich Bullinger expresses his congratulations to Oswald Myconius on recovering his health, hoping it endures so Myconius can continue serving the church and focusing his efforts on it.
A comitiis Badensibus ne hac quidem hora quicquam certi habemus. Legati enim nostri nec redierunt nec literam ulla de re scripserunt. Nihil ergo vel de literis caesaris, quibus de tu scribis, vel de ullis aliis rebus scribere possum.	[2] Die Zürcher Gesandten [Johannes Haab und Itelhans Thumysen], die noch auf der Badener Tagsatzung sind, haben keine Nachricht übermittelt. Bullinger hat also nichts über den von Myconius erwähnten Brief des Kaisers Karl V. gehört.	Bullinger informs Myconius that there has been no certain news from the Baden assembly, as their delegates have neither returned nor written back, leaving him unable to discuss the matters Myconius mentioned or any other issues.
Argentinam vix aliis recipiet in gratiam conditionibus caesar quam alias urbes, quas mellitis verbis et nimbo quodam promissionum occaecat; in fine videbitur cuius.	[3] Straßburg wird kaum unter anderen Bedingungen begnadigt werden als die anderen Städte, die vom Kaiser mit schönen Versprechungen benebelt wurden. Am Ende wird man sehen, von wem [diese Versprechungen ausgingen]!	He speculates that Strasbourg will not easily reconcile with the emperor under conditions different from other cities, which have been blinded by sweet words and promises, indicating a future revelation of true intentions.
...		
Ulma et reliquae miserę urbes exemplum praebent. **Was do̊ͤrffend sich die barmhertzigen lüt also uffzegåͤben? Was nåͤmend sy das tüfelvolck hinyn? Was behaltend sy ir stett nitt unbefleckt und unbeherrschet?** Sed haec est manus domini percutientis nos propter ingratitudinem nostram et vitam poenitere nesciam.	[5] Ulm und die anderen armseligen Städte sind schon ein gutes Beispiel dafür. Warum mussten diese sich ergeben, fremde Besatzungen aufnehmen und ihre Freiheit preisgeben? Bestimmt bestraft Gott so die nicht bußfähigen Menschen für ihre Undankbarkeit.	He discusses the example set by Ulm and other unfortunate cities, questioning the pride of those who accept the deceitful, and lamenting the divine punishment reflected in their plight due to ingratitude and unrepentant living.
...		
Recte iudicas de concordia et discordia caesaris et papae. Ego arbitror saepiae esse atrorem. **Es ist bůbery, damitt die lüt umbgand. Trüw inen ja frylich der tüfel!** Quin potius annunciemus nos regnum dei et salutiferum evangelium filii dei, pontificis et regis nostri, qui syncere agit, neminem fallit et mox venturus est in iustitia, iudicaturus vivos et mortuos. Huic placere in omnibus satagamus! In illo vale cum omnibus bonis. Tiguri, 4. martii circa 9 antemeridianam 1547. Saluta fratres. Bullingerus tuus.	[8] Myconius' Beurteilung von Kaiser und Papst [Paul III.] ist zutreffend. Beide vertuschen nur ihre Kungeleien, um die Menschen besser an der Nase herumzuführen. Umso wichtiger ist es, das Evangelium Christi, des wahren Priesters und treuen Königs, der bald alle richten wird, weiter zu verkündigen! Gruß, auch an die Kollegen. Geschrieben gegen neun Uhr vormittags.	Bullinger critiques the relationship between the emperor and the pope, suggesting it's often more tumultuous than it appears, and advocates for the preaching of God's kingdom and the true gospel, urging to please God in all things. Bullinger concludes with greetings and a personal sign-off, noting the letter's time of writing in Zurich and asking Myconius to greet other brothers.

Excerpt of the human summary (German) vs. the automatic summary (English) of a letter from Heinrich Bullinger to Oswald Myconius, 4. March 1547. Sentences in Early New High German are in bold. See `https://www.bullinger-digital.ch/letter/12884`.

Exploring aspect-based sentiment analysis methodologies for literary-historical research purposes

Tess Dejaeghere[1,4], Pranaydeep Singh[4], Els Lefever[1,4], Julie M. Birkholz[1,2,3]
[1]GhentCDH, Ghent Center for Digital Humanities, Ghent University
[2]Ghent University Department of History
[3]KBR, Royal Library of Belgium, Brussels, Belgium
[4]LT3 - Ghent University Department of Translation, Interpreting and Communication
{tess.dejaeghere, pranaydeep.singh, els.lefever, julie.birkholz}@ugent.be

Abstract

This study explores and compares aspect-based sentiment analysis (ABSA) methodologies for literary-historical research, aiming to overcome the limitations of traditional sentiment analysis in capturing the nuanced aspects of literature. Through the analysis of an English corpus of 19th and 20th-century travelogues, the study develops annotation guidelines and evaluates three ABSA toolchains: a rule-based system, a machine learning-based approach based on both BERT and MacBERTh embeddings, and a prompt-based workflow using the open-source generative large language model Mixtral 8x7B. Findings reveal insights into the challenges and potentials of ABSA methodologies for literary-historical analysis, highlighting the need for context-aware annotation strategies, required technical skills and time investment. The research contributes to the following: (1) the curation of a multilingual corpus comprising 3078 travelogues sourced from online repositories in German, English, French, and Dutch; (2) the publication of an annotated multilingual literary-historical dataset of travelogues for aspect-based sentiment analysis, focusing specifically on environment-related aspects and their associated sentiment scores; (3) creation of openly available and adaptable Jupyter Notebooks with the Python code developed for each modelling approach; (4) publication of pilot experiments for ABSA on literary-historical texts using the English subset of the dataset; and (5) formulation of future endeavors aimed at advancing ABSA methodologies within the realm of literary-historical research.

Keywords: aspect-based sentiment analysis, travelogues, methodology

1. Introduction

The influx of Natural Language Processing (NLP) methodologies in literary-historical research settings remains limited to date (Blevins and Robichaud, 2011; Kuhn, 2019; Kuhn and Reiter, 2015; McGillivray et al., 2020; Suissa et al., 2022). Sentiment analysis (SA) in particular, a popular text mining approach to automatically categorize textual entities as positive, neutral or negative, is critically regarded in literary studies, and often deemed inept to cater to the meticulous research needs of humanist researchers (Kim and Klinger, 2018b; Schmidt and Burghardt, 2018). The reasons for this critique stem largely from the fact that literary analysis can hardly be fit to the inflexible polarity scheme ("positive", "neutral" and "negative") employed by contemporary SA-tools (Buechel et al., 2016; Kim and Klinger, 2018a,b; Kim, 2022; Schmidt and Burghardt, 2018). As a consequence, the application of sentiment analysis in (digital) humanities remains under-explored, and historians and literary scholars are eventually nudged back to a familiar praxis of close reading and manual analysis (Kim and Klinger, 2018b,a; Kuhn, 2019).

1.1. Aspect-based sentiment analysis

To account for the rigid nature of SA-tools, aspect-based sentiment analysis (ABSA) has steadily gained traction. Rather than procuring a polarity label on the level of the document, paragraph or sentence, ABSA systems operate on the aspect-level by combining multiple information extraction subtasks to extract 1) aspect terms 2) aspect categories, 3) opinion terms and 4) sentiment polarities (Birjali et al., 2021; Zhang et al., 2022).

While ABSA is an up-and-coming area of research in NLP, and opening up promising avenues and levels of granularity for sentiment mining, its application currently remains limited to commercial domains such as customer reviews (Zhang et al., 2022). To the knowledge of the authors, the application of ABSA has thus far not been explored for literary-historical textual material.

Unsurprisingly so, perhaps, given that affective patterns in literature often deliberately transcend conventional linguistic structures to translate the enigmatic realm of the intimate human experience (Rebora, 2023). Furthermore, NLP tools are known for introducing their very own implicit (sentiment) theories and biases, contributing an additional stratum of opacity to their application. The rapid advancement of NLP-tools from explainable rule-based systems to models which try to capture abstractions of

LT4HALA 2024@LREC-COLING-2024, pages 129–143

human reasoning further complicates its use in literary analysis contexts, where a demarcation of perspective is paramount. Consequently, a cross-pollination of practices between the two fields is further dwindling, requiring an increasingly intricate set of computational skills and knowledge to build methodological bridges and foster mutual understanding (McGillivray et al., 2020; Rebora, 2023).

The current divide raises the question of whether ABSA as a technique could be a way to circumvent the rigidity of conventional SA-models - granting a more fine-grained and explainable perspective on aspect representation and sentiment expression in literary text. Answering to the calls for exploratory research and evaluation of NLP-based methods, this study presents a pilot endeavor to test a number of ABSA methodologies for literary-historical research contexts (Rebora, 2023).

2. Related work

2.1. Aspect-based sentiment analysis in computational literary studies

In contemporary settings, ABSA is often used in the context of e-commerce to achieve a better understanding of public opinion towards specific aspects of their offered services and products, or to analyze opinions expressed on social media platforms (Mowlaei et al., 2020; D'Aniello et al., 2022; Troya et al., 2022). While sentiment categories are usually constrained to a five- or three-point scale – previous work explored fine-grained emotion categories tied to an aspect to improve customer relation management (De Geyndt et al., 2022).

Literature on ABSA is characterized by its scattered nature, and the scientific terminology employed to delineate this task lacks uniformity. While "aspect-based sentiment analysis" is largely accepted as the standard nomenclature – the task has been referred to as ACOD (aspect-category-opinion-sentiment quadruple extraction), TOWE (target-oriented opinion word extraction) (Xu et al., 2020), ELSA (entity-level sentiment analysis) (Rønningstad et al., 2023), TSA (targeted sentiment analysis) (Zhang et al., 2016), ASAP (aspect category sentiment analysis and rating prediction) (Bu et al., 2021) among a myriad of other denominations. Indeed, "the terminologies of ABSA studies are often used interchangeably, but sometimes they have different meanings according to the context [...] This may cause unnecessary confusion and often makes the literature review incomplete (Zhang et al., 2022).". Next to the mere terminological nature of this debate – what is defined as an aspect and a sentiment differs across applications and "must be treated using completely different approaches as they lead to different kind of results" (D'Aniello et al., 2022). While this fuzzy use of terminology is likely not the primary impediment to the adoption of this technique in DH settings – it may further obscure the definition of the methodology itself and the necessary distinct subtasks involved, posing an additional hurdle for scholars less familiar with NLP jargon when attempting to integrate this technique or assess its application range.

Depending on the desired output, different learning strategies are combined for the aspect recognition and sentiment analysis subtasks respectively – ranging from unsupervised (e.g.: frequency, statistics, heuristics, dependency parsing, rule-based approaches, zero-shot classification or topic modelling, etc.), semi-supervised (e.g.: lexicons and lexicon generation, dependency trees or knowledge graphs, etc.), and supervised (e.g.: machine learning, decision trees, neural networks, etc.) strategies (Birjali et al., 2021; D'Aniello et al., 2022; Keshavarz and Abadeh, 2017; Pattakos, 2021; Xu et al., 2021; Zhang et al., 2022). In more recent work, the power of generative language models for zero-shot and few-shot classification were also explored (Hosseini-Asl et al., 2022; Pangrazzi, 2022; Vector Institute, 2023).

Considering the traction gained by tasks such as Named entity recognition (NER), relation extraction (REX) and sentiment analysis (SA) in humanist research (Al-Razgan et al., 2021; Arnoult et al., 2021; Gamallo and Garcia, 2019; Jänicke et al., 2017; Li, 2022; Neudecker, 2016; Pineda et al., 2020; Todorov and Colavizza, 2020; Won et al., 2018) – it is but a small step to envision the potential of ABSA, which amalgamates the capabilities of these individual techniques. Apart from recent work which compares the application of ChatGPT to an in-house fine-tuned BERT architecture applied to a set of literary reviews (Martens et al., 2023) – ABSA has not yet been applied within the domain of computational literary studies.

While positing ABSA as a panacea would be a gross exaggeration, trying new methodologies to assess the applicability of NLP in DH practice is paramount. Rather than presenting a full-fledged solution, this study aims to answer to the calls for an exploratory approach in NLP-infused literary analysis methodologies, guided by the principle that "a criticism of the tools and methods currently adopted in sentiment analysis is as necessary as a free exploration of its potential (Rebora, 2023)".

2.2. Annotation and evaluation

While the annotation process is widely considered essential for the development and evaluation of information extraction tasks, literary texts are known to be extraordinarily tedious and difficult to annotate due to their subjective nature and stylistic properties (Kleymann and Stange, 2021; Ivanova et al., 2022; Ehrmann et al., 2021). Figurative language such as metaphors, personification and metonymy; stylistic and language-specific peculiarities across authors' works and the specific research needs of literary scholars and historians hamper a standardisation of annotation practices across the entire literary domain (Bamman et al., 2019). Additionally, the historical variety space in which a text resides further obfuscates its interpretation and, therefore, the annotation process for targeted information extraction tasks (Plank, 2022).

Despite previous attempts at the creation of annotated datasets and annotation frameworks for NER within the domain of English literature by for example LitBank (Bamman et al., 2019) and the calls for targeted approaches and "agreed-upon annotation guidelines to be used for the annotation of literary novels (Ivanova et al., 2022)" - the highly individual text analysis needs of literary scholars and historians require a more flexible approach (D'Aniello et al., 2022; Jacobs, 2019; McGillivray et al., 2020).

Regarding evaluation, utilizing or merging existing datasets to serve as a benchmark representative of the "literary data" domain has not yielded fruitful results. Because of the wide variety of annotation practices and the diverse characteristics featured across these test sets, using different partitions of the gold standard annotations may lead to vastly different evaluation outputs (Ivanova et al., 2022). Additionally, NLP-native evaluation metrics such as accuracy and F1 scores often do not cater to the meticulous evaluation practices in the humanities - thus making annotation and evaluation "[...] all the more challenging as the scope of needs and applications in humanities research is much broader than the one usually addressed in modern NLP (Ehrmann et al., 2021)" (Klinger et al., 2020; Rebora, 2023).

3. Methodology

3.1. Travelogues as data

As a use-case to test these methodologies, attention is geared towards the application of ABSA to a textual corpus of travelogues from the 19th and 20th centuries.

Travelogues are an extraordinarily interesting source in this respect - as they constitute an idiosyncratic lens on the author's travel experiences - thus granting readers an intimate glimpse into the writer's identity and views on their surroundings (Colletta et al., 2015; José and Joseph Parathara, 2018; Sprugnoli, 2018). Leveraging this unique characteristic, the study zooms in on a set of aspects related to the environment as perceived and documented in the travelogue. Not only standard aspects such as people, locations, organizations are annotated, but we further enriched the data with aspect annotations related to weather phenomena, natural landforms, human landforms, biomes, fauna, and flora. While beyond the current study's scope, the resulting open-source dataset could serve as a catalyst to foster a more profound understanding of the historical value attributed to nature through literary analysis, or as a benchmark dataset for future ABSA methodologies in the literary-historical domain (Virdis, 2023; Correia et al., 2021; Langer et al., 2021; van Erp et al., 2018).

1. **Dataset collection**: as a first step, the collection of a multilingual corpus comprising of 3078 non-fictional travelogues from the 19th to the 20th century in English, French, Dutch and German from a range of online repositories is described.

2. **The development of annotation guidelines** tailored to the annotation of aspects and sentiments in travelogues is explained, as well as the selection of annotators. As a proof of concept, a subset of the corpus consisting of 58 texts across languages is subjected to annotation according to these guidelines by three trained student annotators.

3.2. ABSA pipeline development

The development and evaluation of three ABSA-pipelines, one supervised system and two unsupervised systems, is further detailed.

1. **A rule-based system** is developed for 1) aspect extraction based on spaCy's noun extraction module, 2) opinion word identification using spaCy's POS-tagger to extract adjectives, adverbs and auxiliary constructions and 3) sentiment analysis based on the extracted opinion words using the SenticNet package. In the case of negated sentiment words, NLTK's synset module was used to fetch the word's antonym and generate a score (Loper and Bird, 2002; Cambria et al., 2020; Montani et al., 2023).

2. **A machine learning-based** pipeline is developed in two steps. The aspect extraction task is tackled by training two Flair-based

sequence taggers on the annotations. One of the sequence taggers is based on BERT embeddings, while the other is trained using MacBERTh embeddings. Their performances are evaluated on the gold standard aspects using 5-fold cross-validation, and compared. For the sentiment analysis task, BERT and MacBERTh models were fine-tuned on the gold standard aspects. These embeddings subsequently serve as input for diverse machine learning classification architectures, including SVM, AdaBoost, Random Forest, and MLP classifiers (Devlin et al., 2019; Manjavacas Arevalo and Fonteyn, 2021; Greve et al., 2021).

3. **A prompt-based zero-shot workflow** using the multilingual generative Large Language Model Mixtral-8x7B-Instruct-v0.1 is developed. Experiments with prompts, parameter settings and output parsing steps are discussed for the aspect and sentiment extraction tasks respectively (Jiang et al., 2024).

Our developed methodologies are compared in terms of time investment, required expertise, and the level of transparency and usability for humanist research purposes. The final evaluation is conducted from a methodological point of view, and not geared towards the improvement or comparison of model performances. Furthermore, we evaluate the suitability of the ABSA approaches for the literary-historical domain and propose directions for future research.

4. Results and discussion

4.1. Data gathering

The travelogues feature diverse genres such as nature writing, travel memoirs, journals, and poetry. It must also be acknowledged that a non-fictional nature of these texts cannot be fully assumed – as these stories are often, though not always, a concoction of fact and fiction. The documents were sourced from various online repositories as outlined below, and resulted in a dataset of 3,320 texts across the languages English, French, Dutch and German as shown in Table 1. Opposite to the other collections, the texts gathered from the Biodiversity Heritage Library as well as those fetched from the Travelogues project included OCR-related mistakes. Using the garbageness score as a quality filter, the most extreme cases were filtered out (Ryan, 2015).

1. Travel-related texts from the Biodiversity Heritage Library[1] were scraped via API using travel-related terms and primarily feature non-fictional travel reports by biologists and naturalists .

2. The subcollection sourced from DBNL (Digitale Bibliotheek voor Nederlandse Letteren) [2] consists mainly of Dutch stories and reports on colonial explorations by Dutch-speaking settlers.

3. Italian travel reports comprise narratives about Italy written by English authors in the 1930s (Sprugnoli, 2017).

4. The Arctic Travellers dataset was manually collected from the Internet Archive[3].

5. Non-fictional travel reports were gathered from Project Gutenberg[4].

6. A set of German travelogues from the Travelogues project, available for download on their GitHub repository, were automatically compiled by domain experts (Rörden et al., 2020)[5].

Language	**18thC**	**19thC**	**20thC**	**Total**
English	41	782	668	**1,491**
French	5	145	50	**200**
Dutch	25	92	242	**359**
German	972	218	80	**1,270**
Total	**1,043**	**1,163**	**897**	**3,320**

Table 1: Overview of languages contained in the travelogues corpus (approx. 5,000 tokens/text)

Finally, 58 texts were annotated across all the languages present in the corpus (English, French, Dutch and German) using the platform INCEPTION (Klie et al., 2018). As a proof of concept, this work focuses on the English subset of this gold standard data. This is a subset of 22 texts. After training the students to use the annotation platform and the annotation guidelines, 14 texts of approximately 500 tokens each were annotated by all annotators to calculate the inter-annotator agreement (Fleiss' kappa score) for the aspect categories and the sentiment annotation on both aspect and sentence levels as shown in Table 2. Interestingly, these results indicate that students found it more difficult to annotate sentiment on the level of the sentence than on the level of the aspect. This may be because it is simply harder to assess the sentimental value of an entire sentence. While the Kappa score for the aspect categories PERSON, LOCATION,

[1] https://www.biodiversitylibrary.org/

[2] https://www.dbnl.org/

[3] https://www.archive.org/

[4] https://www.gutenberg.org/

[5] https://www.travelogues-project.info/

ORGANISATION, FAUNA, FLORA, BIOME, HUMAN_LANDFORM, NATURAL_LANDFORM, NATURAL_PHENOMENON, WEATHER, MYTH and BIOME was quite high, categorization of these aspects is not the focus of this work.

Annotation	Kappa
Aspect category	0.88
Sentiment (aspect)	0.64
Sentiment (sentence)	0.24

Table 2: Overview of the inter-annotator agreement Fleiss' Kappa scores across sentiment and aspect annotations for English

4.2. Annotation process

Student annotators were chosen based on their language proficiency across the languages featured in the corpus. The students were working on studies in history or multilingual communication. At all times, with the exception of the annotations used to calculate the IAA, the students were allowed to engage in discussions with one another to foster an exchange of historical and linguistic expertise. The texts' metadata was released to the students and included information on release dates, full titles and authors, allowing them to look up more contextual information if needed. Discussions regarding recurring ambiguous aspect categories often spontaneously took on a rather philosophical nature (e.g.: should we indicate "God" as a PERSON or MYTH aspect?), and decisions were gradually adjusted depending on the cases encountered. Metaphors also regularly surfaced (e.g.: "Eternal City" as a denomination for "Rome") and annotated.

Because we attempt to model the readers' evaluative response to the text rather than the intended sentiment value of the author, the students were asked to annotate sentiment based on their own affective evaluation of the text. Given the unpredictable shape of literary text, the only rule implemented to distinguish between the extreme categories 1 and 5 was the presence of intensifiers in the chunk (adverbs such as "very" or "extremely"). It quickly became evident during our discussions that the five-point scale for sentiment introduced too much ambiguity, and during the modelling phase it was decided to compress the categories to a three-point scale and compare performances. Examples of ambiguous cases are legion and their thorough discussion could easily be the subject of separate research efforts. One example is shown in Figure 1, where a colonial traveller discusses an encounter with the indigenous Indian population and refers to them in his travelogue by describing them as "civilised". Our annotator deemed this a positive expression connected to the aspect "Indians", but given the colonial context in which this text was written, the need of the author to explicitly mention the "civilised" nature of these people expresses a level of surprise, harbouring a condescending and thus negative depiction of the aspect indians through a contemporary reader's lens.

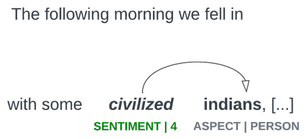


Figure 1: Example of ambiguous sentiment annotation.

Another example depicted in Figure 2 showcases the layered sentimental expression often present in literary sentences, and underlines the usefulness of ABSA as a fine-grained methodology.

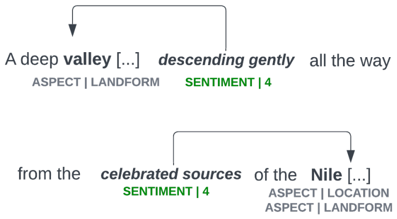


Figure 2: Example of the annotation of layered sentimental expression in a single sentence.

4.3. Aspect extraction

Our annotations were converted to a BIO-format, and the output of our aspect extraction models was evaluated using the nervaluate package[6] and a strict macro F1 approach and shown in Table 3.

4.3.1. Unsupervised approaches

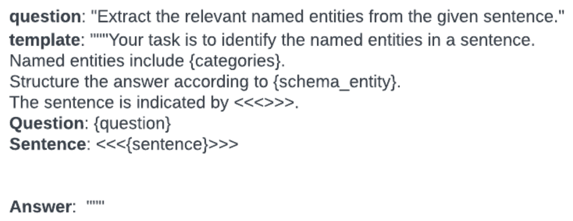
Our rule-based system constituted a simple approach which follows the notion of noun chunks as optimal aspect candidates, while adjectives and adverbs serve as potential opinion words - as suggested by previous work (Anwar et al., 2023; Nandhini et al., 2018; Mai and Zhang, 2020; Nandhini et al., 2018; Anwar et al., 2023). SpaCy was used to extract nouns and proper nouns from the noun chunks which were then converted to BIO-format and evaluated against the annotations. The discrepancy of our annotations and this rather one-dimensional approach is reflected in the low strict

[6] https://pypi.org/project/nervaluate/

F1 scores (0.20). A manual analysis of the errors showed that the rule-based system's mistakes are logically mostly due to extraction of irrelevant nouns as aspects (e.g.: "Sunday", "a brief visit", "lower end", "ugliness", "unusual distance") which are not part of the categories under consideration. Conversely, in some cases, the approach revealed entities that were missed by the annotators.

The other unsupervised system using the generative Mixtral-8x7B-Instruct-v0.1 model, was implemented through the LangChain development framework as a zero-shot approach (Harrison, 2022). Being a recently developed technology at the time of writing, pitfalls and strengths of these generative LLMs across domains are yet to be discovered. The biggest challenge for a digital humanist to overcome here is not necessarily producing the code itself, but finding the correct way of constructing a prompt of which the output can be consistently parsed while retaining awareness of the model's inherent bias and tendencies to hallucinate. Using a development set of annotated samples as input texts, we experimented with the temperature setting, which was eventually set to the low value of 0.01 as this intuitively renders the least convoluted results. To make the output easy to parse, we designed a JSON output schema as the example shown in Figure 5 and asked the model to generate the output accordingly. Without this structural element, the model's output was unstructured and consequently impossible to parse consistently.

Categories were added as context information as a string object, and included a short definition for each category between brackets. The input sentence was indicated in the prompt using designated symbols to ensure the model relies solely on the input sentence to construct an answer. The finalized prompt is shown in Figure 3. A clear task description ("Extract the relevant named entities from the given sentence") was used as input. Upon experimentation, it became clear that the model produced better results when asked to extract "named entities" as compared to "aspects". It was noted that in both cases, the model extracted common names, personal pronouns ("he", "her") as well as proper names, which may need to be tweaked through the prompt depending on the use-case. Interestingly, the concept of "location" was quite literally interpreted by the model, and snippets such as "convenient places", "over there" and "the latter place" were also extracted. We experimented with adding a personality to the model (e.g.: "You are a historian and literary scholar with expertise on historical travel literature"). Interestingly, adding this feature sometimes caused the model to add an unrequested lengthy explanation about its reasoning, a feature which could be useful for humanists to adjust their prompting techniques and decide on which contextual information and examples to add in a few-shot setting, allowing for an adequate and intuitive human-in-the-loop setting.



```
question: "Extract the relevant named entities from the given sentence."
template: """Your task is to identify the named entities in a sentence.
Named entities include {categories}.
Structure the answer according to {schema_entity}.
The sentence is indicated by <<<>>>.
Question: {question}
Sentence: <<<{sentence}>>>


Answer: """
```

Figure 3: Prompt for aspect extraction

While the results of this approach on a held-out test set are not high (0.34 F1), the output was still impressive considering the limited contextual information that was given in the prompt, and warrants further research in this domain.

4.3.2. Supervised approach

Our machine-learning based aspect extraction model was made using Flair's SequenceTagger module, and evaluated through 5-fold cross-validation on equal splits of the data. BERT- and MacBERth-embeddings were used respectively to train two different taggers. Surprisingly, the BERT embeddings in this case rendered a better macro F1 score (0.62) and trumped the MacBERTh embeddings made for historical English (0.59). The code for this operation was easy to retrieve and adapt through Flair's documentation, but does require a basic understanding of embeddings and parameter settings.

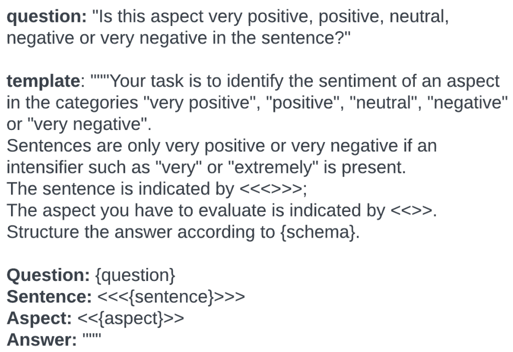
Unsupervised models	**F1**
Rule-based system	0.20
Mixtral 8x7b	0.34
Supervised models	
SQT Flair BERT	**0.62**
SQT Flair MacBERTh	0.59

Table 3: Overview of scores for the English aspect extraction models on a test set

4.4. Sentiment analysis

4.4.1. Unsupervised approaches

For the rule-based system, we wanted to evaluate the system on the text snippets that were labelled with a sentiment and connected to an aspect in the gold standard data. Thus, we had to look for language-specific tools that were able to output a sentiment score based on a given text chunk. For English, luckily, quite a few lexicon-based tools are available for sentiment analysis. Eventually, the tool Senticnet was applied and evaluated on the opinion words in the annotated data (Cambria et al., 2020). This tool was chosen for its ease of use and transparency in terms of the used emotion ontology and polarity scoring principles. Using a sigmoid function, the resulting float scores returned by Senticnet $\in [-1:1]$ were normalized into a [0:1] float range for each sentiment-bearing word. The final "sentiment score" is the mean of the scores for each word. After that, a threshold was determined and linked to a respective sentiment label (if the mean score is equal to or less than 0.20, the sentiment label is 1; if the score is equal to or similar to 0.40, the sentiment label is 2 and so forth) to match the range of the annotations. Negations occurring in the noun phrases (e.g.: "*not* beautiful") were addressed by finding the antonym of the negated word using NLTK's synsets module - and then applying SenticNet to the fetched antonym (Loper and Bird, 2002). Intensifiers, given that these were explicitly mentioned in the annotation guidelines and are thus expected to influence the annotations, were considered by checking whether an adverb is present in the noun phrase, and pushing the mean score into category 1 if it's below or equal to the 0.50 threshold, or 5 if it's above. As shown in 4, the system consistently performed better when compressing the scoring system in the 1-3 range. The packages used, while multilingual, are not tailored to historical language, which was not a serious shortcoming for English, but undoubtedly would be in the case of lesser-resourced (historical) languages, which makes this approach less advisable in most cases.

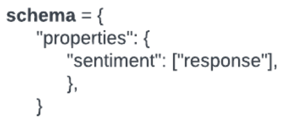


question: "Is this aspect very positive, positive, neutral, negative or very negative in the sentence?"

template: """Your task is to identify the sentiment of an aspect in the categories "very positive", "positive", "neutral", "negative" or "very negative".
Sentences are only very positive or very negative if an intensifier such as "very" or "extremely" is present.
The sentence is indicated by <<<>>>;
The aspect you have to evaluate is indicated by <<>>.
Structure the answer according to {schema}.

Question: {question}
Sentence: <<<{sentence}>>>
Aspect: <<{aspect}>>
Answer: """

Figure 4: Prompt for sentiment analysis using Mixtral 8x7b



```
schema = {
    "properties": {
        "sentiment": ["response"],
        },
    }
```

Figure 5: Output JSON schema for sentiment analysis using Mixtral 8x7b

The prompt for the Mixtral 8x7b was constructed much in the same way as that of the aspect extraction. A clear indication of the sentence under consideration and an expected output structure as shown in Figure 5 was confirmed to be really important. Here, too, the personality addition ("you are a historian") made for a more convoluted output and produced a string of reasoning, which made the output unpredictable and difficult to parse, but was interesting to further scrutinize. In one example, the aspect "officers" in the sentence "[...] he, accordingly to a plan long since proposed , formed the Indians into Companies and by degrees taught them to feel the convenience of having officers set apart to each , which they were soon not only reconciled to but highly pleased with , by which means he gave some degree of method and form to the most Independent race of the Indians [...]", was positively evaluated, because, according to the model: "The sentence expresses that the officers were able to teach the Indians to feel the convenience of having officers set apart to each, which they were soon not only reconciled to but highly pleased with. This implies that the officers were able to positively influence the Indians and make them feel more organized and structured.", echoing a contextless and historically unnuanced assessment of the text material which may be considered dangerously biased in research contexts.

Unsupervised models	**F1**
Rule-based system (1-5)	0.32
Rule-based system (1-3)	**0.37**
Mixtral 8x7b (1-5)	0.33
Mixtral 8x7b (1-3)	**0.42**

Table 4: Overview of unsupervised sentiment model scores for English

4.4.2. Supervised approaches

Our approach was adapted from previous work by Greve et al. (2021), which trained embeddings using BERT and used them as features in machine learning models to differentiate between positive and negative literary reviews. BERT and MacBERTh embeddings were trained for sentiment labels on a 1-5 point scale and labels on a 1-3 point scale respectively, and used as input for a variety of ML-models (SVM, MLP, RF and AdaBoost). The MLP classifier, a Multi-Layer Perceptron classifier, consistently outperformed the other networks as shown in Table 5.

Embeddings	Model	F1
BERT (1-5)	SVM	0.53
	MLP	**0.56**
	RF	0.49
	AdaBoost	0.42
MacBERTh (1-5)	SVM	0.55
	MLP	**0.57**
	RF	0.49
	AdaBoost	0.43
BERT (1-3)	SVM	0.60
	MLP	**0.61**
	RF	0.50
	AdaBoost	0.50
MacBERTh (1-3)	SVM	0.57
	MLP	**0.62**
	RF	0.51
	AdaBoost	0.49

Table 5: Overview of supervised sentiment model scores for English

4.5. Qualitative comparison of methodologies

Designing the rule-based model was a time-consuming process, and requires not only thorough knowledge of the content of the data, but also of the linguistic manifestation of sought-after information. While most corpora for literary-historical use-cases are indeed limited in size, nouns phrases are, as expected, unfit to uncover complicated literary vehicles such as metaphors and simile, which may skew results. Additionally, sentiment lexica and tools for historical vernaculars were hard to find for the English language domain, let alone for other lesser-resourced languages, which would be a considerable impediment for developing a rule-based system in most DH research settings. However, the transparency of this white-box approach does grant the user a sense of control over the output, and does not require a thorough knowledge of modelling practices. Summarized, this approach seems advisable in the case of small corpora and cases where the grammatical structure of the aspects to be extracted is known and relevant to the use-case, or where sentiments are expressed using predictable words and formulae.

In the case of the generative model Mixtral, it was noted that the model sometimes had a tendency to hallucinate aspect categories that were not given in the prompt. Depending on how the prompt is formulated, the output included unrequested information beyond defined aspects or sentiment categories, and was sometimes unpredictable in shape and thus difficult to parse. Additionally, how a sentiment value is calculated exactly based on the input sentence is not clear, and one must keep into account that even this output may be no more than a model's best guess. From a technical point of view, this approach is quickly gaining traction at the time of writing, as many new open- and closed-source models and prompting techniques are being developed. This oversupply could make it challenging for the humanist researcher to find a fitting and well-documented generative open-source approach for a specific use-case. Multiple existing frameworks and models are currently behind a pay-wall, which raises questions regarding the privacy of research output and impedes widespread use. The open-source models through HuggingFace have installed a limit on server requests, which should be taken into account when planning to apply this methodology to large datasets. Indeed, it is possible to use these models to produce output quite easily, even with a basic understanding of the inner workings of generative LLM and programming, which makes them an attractive option for information extraction, but a thorough evaluation of its output is advised.

Machine learning and deep learning approaches have been favoured in computational literary-historical settings in the last couple of years. Logically, fine-tuning these systems creates output which remains more faithful to the annotations than the rule-based or generative model-based approaches, making it a reliable methodology in this context. Adapting existing code or creating new systems requires at least basic background understanding of embeddings, Tensor operations and a meta-understanding of neural networks and modelling using the HuggingFace platform. For digital humanists without this knowledge of NLP-practices, adapting and implementing this code may be too time-consuming. Additionally, machine learning models are data-hungry, and the effort required to produce annotations and enable training may be disproportionate if its application will be limited to a small case-study.

4.6. Contributions

This study makes the following contributions that includes the sharing of annotated data to knowledge on these practices:

1. A novel multilingual dataset of 3,320 travelogues ranging from the 19Th until the 20Th centuries is gathered from a range of online sources and made public on our GitHub repository [7].

2. Insights are formulated on the creation of annotation guidelines and their application to the literary-historical domain.

3. The annotated subset of this dataset for aspect-based sentiment analysis in English, Dutch,

[7] https://github.com/TessDejaeghere/Travelogues

German and French as well as the annotation guidelines used are made open-source, encouraging reuse for further research endeavours in the domain of aspect-based sentiment analysis and literary-historical research on travelogues.

4. We introduce pioneering work on the assessment of aspect-based sentiment analysis methodologies for the domain of computational literary studies, ranging from white box rule-based techniques to state-of-the-art black box techniques using a generative LLM. The code developed for this research is made open-source in the form of annotated Jupyter Notebooks to facilitate adaptability and reuse by computational linguists and (digital) humanists alike.

5. Conclusion and future research

The research explored three methodologies for ABSA in literary-historical research contexts. First and foremost, it must be noted that annotating biodiversity in travelogues is a fully-fledged research project an sich. Annotating literary-historical texts for research purposes is exceptionally challenging, and as opposed to the breadth-oriented approach in contemporary NLP settings, Digital Humanists can hardly escape the depth-oriented strategy to cater to their meticulous needs. Rather than adopting an exploratory lens, it is advised to use these information extraction techniques for well-defined research ends, and the availability of sufficient data and time should warrant its development (Chun and Elkins, 2023). The methodologies assessed come with their unique set of advantages and disadvantages: in the absence of sufficient annotated material, a large corpus as a use-case or knowledge of NLP practices, machine learning approaches may oftentimes not merit the effort. Rule-based systems do not require this knowledge of NLP-techniques and may work well in settings where the aspect and the sentiment expressions follow strict and formulaic patterns, but are often time-consuming to create. Unlike the level of expertise required for ML approaches, the methodology involving prompting the generative LLM Mixtral 8x7b is fairly straightforward. However, one must tread with great care when applying this methodology for literary-historical research applications, as our experiments confirmed the tendency of these models to hallucinate unrequested information. Additionally, specifically in the case of sentiment analysis, it is unclear how the engine makes its assessment. At the time of writing, a myriad of new models are created on a daily basis, which makes choosing an adequate model rather challenging. Researchers should also be aware of privacy concerns when using closed models versus open-source models on their dataset. However, generative LLMs could present an exciting new way to answer the call for a grey-box human-in-the-loop approach, but further research is needed to explore pitfalls and possible evaluation schemes:

- Future research may delve into the implementation of ABSA within a case-study framework, juxtaposed with a manual methodology for comparison.
- Using the novel multilingual travelogues dataset annotated for ABSA presented in our research, we aim to gear our future efforts towards methodological research expansion in sentiment analysis, NER and ABSA across diverse linguistic and literary-historical contexts. Future research endeavors might be directed towards the development of novel evaluation methodologies that transcend the conventional metrics employed in NLP. Such inquiries could contemplate whether outputs divergent from gold standard data necessarily constitute inaccuracies, or if they offer alternative perspectives that could augment human assessment.
- Further exploration into generative models across varied contexts presents an intriguing avenue. This includes investigating the impact of bias and model hallucinations on information extraction tasks like ABSA, as well as experimenting with different prompting techniques, incorporating contextual information, and even diverse modalities. Such endeavors could establish the groundwork for a human-in-the-loop grey-box evaluation methodology, wherein researchers engage in dialogue with the corpus, assess output samples, and adapt prompts accordingly.

6. Bibliographical References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Muna Al-Razgan, Asma Alrowily, Rawan N. Al-Matham, Khulood M. Alghamdi, Maha Shaabi, and Lama Alssum. 2021. Using diffusion of innovation theory and sentiment analysis to analyze attitudes toward driving adoption by Saudi women. *Technology in Society*, 65:101558.

Muchamad Anwar, Dedy Trisanto, Ahmad Juniar, and Fitra Sase. 2023. Aspect-based Sentiment Analysis on Car Reviews Using SpaCy Dependency Parsing and VADER. *Advance Sustainable Science Engineering and Technology*, 5:0230109.

Sophie I. Arnoult, Lodewijk Petram, and Piek Vossen. 2021. Batavia asked for advice. Pretrained language models for Named Entity Recognition in historical texts. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 21–30, Punta Cana, Dominican Republic (online). Association for Computational Linguistics.

Lidia Bocanegra Barbecho. 2023. Review: BERT for Humanists. *Reviews in Digital Humanities*, IV(4).

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*, 1st ed edition. O'Reilly, Beijing ; Cambridge [Mass.]. OCLC: ocn301885973.

Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane. 2021. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226:107134.

Cameron Blevins and Andrew Robichaud. 2011. 2: A Brief History » Tooling Up for Digital Humanities.

Jiahao Bu, Lei Ren, Shuang Zheng, Yang Yang, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. ASAP: A Chinese Review Dataset Towards Aspect Category Sentiment Analysis and Rating Prediction. ArXiv:2103.06605 [cs].

Sven Buechel, Johannes Hellrich, and Udo Hahn. 2016. Feelings from the Past—Adapting affective lexicons for historical emotion analysis. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 54–61, Osaka, Japan. The COLING 2016 Organizing Committee.

Lawrence Buell, Ursula K. Heise, and Karen Thornber. 2011. Literature and Environment. *Annual Review of Environment and Resources*, 36(1):417–440.

Jon Chun and Katherine Elkins. 2023. eXplainable AI with GPT4 for story analysis and generation: A novel framework for diachronic sentiment analysis. *International Journal of Digital Humanities*, 5(2):507–532.

L. Colletta, J. Buzard, C. Chard, C.E. Hornsby, L. Olcelli, S. Russell, N. Stanley-Price, J. Suh, and A. Thompson. 2015. *The Legacy of the Grand Tour: New Essays on Travel, Literature, and Culture*. Fairleigh Dickinson University Press.

Ricardo A. Correia, Richard Ladle, Ivan Jarić, Ana C. M. Malhado, John C. Mittermeier, Uri Roll, Andrea Soriano-Redondo, Diogo Veríssimo, Christoph Fink, Anna Hausmann, Jhonatan Guedes-Santos, Reut Vardi, and Enrico Di Minin. 2021. Digital data sources and methods for conservation culturomics. *Conservation Biology*, 35(2):398–411.

Michael Cronin. 2022. *Eco-Travel: Journeying in the Age of the Anthropocene*. Elements in Travel Writing. Cambridge University Press.

Ellen De Geyndt, Orphee De Clercq, Cynthia Van Hee, Els Lefever, Pranaydeep Singh, Olivier Parent, and Veronique Hoste. 2022. SentEMO: A multilingual adaptive platform for aspect-based sentiment and emotion analysis. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 51–61, Dublin, Ireland. Association for Computational Linguistics.

Giuseppe D'Aniello, Matteo Gaeta, and Ilaria La Rocca. 2022. KnowMIS-ABSA: an overview and a reference model for applications of sentiment analysis and aspect-based sentiment analysis. *Artificial Intelligence Review*, 55(7):5543–5574.

Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. 2021. Named Entity Recognition and Classification on Historical Documents: A Survey. *arXiv:2109.11406 [cs]*. ArXiv: 2109.11406.

Martin Paul Eve. 2022. Distance and History. In Martin Paul Eve, editor, *The Digital Humanities and Literary Studies*, page 0. Oxford University Press.

Justyna Fruzińska. 2022. *Nineteenth-century visions of race: British travel writing about America*. Routledge studies in nineteenth century literature. Routledge, Taylor & Francis Group, New York.

Pablo Gamallo and Marcos Garcia. 2019. Editorial for the Special Issue on “Natural Language Processing and Text Mining”. *Information*, 10(9):279.

Lore De Greve, Pranaydeep Singh, Cynthia Van Hee, Els Lefever, and Gunther Martens. 2021. Aspect-based Sentiment Analysis for German: Analyzing “Talk of Literature” Surrounding Literary Prizes on Social Media. *Computational Linguistics in the Netherlands Journal*, 11:85–104.

Jo Guldi. 2023. Why Textual Data from the Past Is Dangerous. In *The Dangerous Art of Text Mining: A Methodology for Digital History*, pages 25–56. Cambridge University Press, Cambridge.

Andrew J. Hansen, Ruth S. DeFries, and Woody Turner. 2012. Land Use Change and Biodiversity: A Synthesis of Rates and Consequences during the Period of Satellite Imagery. In Freek D. Van Der Meer, Garik Gutman, Anthony C. Janetos, Christopher O. Justice, Emilio F. Moran, John F. Mustard, Ronald R. Rindfuss, David Skole, Billy Lee Turner, and Mark A. Cochrane, editors, *Land Change Science*, volume 6, pages 277–299. Springer Netherlands, Dordrecht. Series Title: Remote Sensing and Digital Image Processing.

Daniel Hershcovich, Stella Frank, Heather Lent, Miryam de Lhoneux, Mostafa Abdou, Stephanie Brandl, Emanuele Bugliarello, Laura Cabello Piqueras, Ilias Chalkidis, Ruixiang Cui, Constanza Fierro, Katerina Margatina, Phillip Rust, and Anders Søgaard. 2022. Challenges and Strategies in Cross-Cultural NLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6997–7013, Dublin, Ireland. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Wenhao Liu, and Caiming Xiong. 2022. A Generative Language Model for Few-shot Aspect-Based Sentiment Analysis. ArXiv:2204.05356 [cs].

Rositsa Ivanova, Marieke van Erp, and Sabrina Kirrane. 2022. Comparing Annotated Datasets for Named Entity Recognition in English Literature. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3788–3797, Marseille, France. European Language Resources Association.

Arthur M. Jacobs. 2019. Sentiment Analysis for Words and Fiction Characters From the Perspective of Computational (Neuro-)Poetics. *Frontiers in Robotics and AI*, 6.

George José and Thomas Joseph Parathara. 2018. Travelogues: Concept and The Text. *International Journal of Creative Research Thoughts*, 6(2).

Dan Jurafsky and James H. Martin. 2023. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*, 3rd ed edition. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, Upper Saddle River, N.J. OCLC: 213375806.

S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann. 2017. Visual Text Analysis in Digital Humanities. *Computer Graphics Forum*, 36(6):226–250.

Selin Kesebir and Pelin Kesebir. 2017. A Growing Disconnection From Nature Is Evident in Cultural Products. *Perspectives on Psychological Science*, 12(2):258–269.

Hamidreza Keshavarz and Mohammad Saniee Abadeh. 2017. ALGA: Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs. *Knowledge-Based Systems*, 122:1–16.

Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2023. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, 82(3):3713–3744.

Evgeny Kim and Roman Klinger. 2018a. A survey on sentiment and emotion analysis for computational literary studies. *CoRR*, abs/1808.03137.

Evgeny Kim and Roman Klinger. 2018b. Who Feels What and Why? Annotation of a Literature Corpus with Semantic Roles of Emotions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1345–1359, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Hoyeol Kim. 2022. Sentiment Analysis: Limits and Progress of the Syuzhet Package and Its Lexicons. *DHQ: Digital Humanities Quarterly:*, 16(2).

Hannah Rose Kirk, Yennie Jun, Filippo Volpin, Haider Iqbal, Elias Benussi, Frederic Dreyer, Aleksandar Shtedritski, and Yuki Asano. 2021.

Bias Out-of-the-Box: An Empirical Analysis of Intersectional Occupational Biases in Popular Generative Language Models. In *Advances in Neural Information Processing Systems*, volume 34, pages 2611–2624. Curran Associates, Inc.

Rabea Kleymann and Jan-Erik Stange. 2021. Towards Hermeneutic Visualization in Digital Literary Studies. *DHQ: Digital Humanities Quarterly*, 2(15).

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics.

Roman Klinger, Evgeny Kim, and Sebastian Padó. 2020. Emotion Analysis for Literary Studies: Corpus Creation and Computational Modelling. In *Emotion Analysis for Literary Studies: Corpus Creation and Computational Modelling*, pages 237–268. De Gruyter.

Jonas Kuhn. 2019. Computational text analysis within the Humanities: How to combine working practices from the contributing fields? *Language Resources and Evaluation*, 53(4):565–602.

Jonas Kuhn and Nils Reiter. 2015. *A Plea for a Method-Driven Agenda in the Digital Humanities* . Sydney, Australia.

Lars Langer, Manuel Burghardt, Roland Borgards, Katrin Böhning-Gaese, Ralf Seppelt, and Christian Wirth. 2021. The rise and fall of biodiversity in literature: A comprehensive quantification of historical changes in the use of vernacular labels for biological taxa in Western creative literature. *People and Nature*, 3(5):1093–1109.

Nicolas Le Guillarme and Wilfried Thuiller. 2022. TaxoNERD: Deep neural models for the recognition of taxonomic entities in the ecological and evolutionary literature. *Methods in Ecology and Evolution*, 13(3):625–641.

Jingxia Li. 2022. Emotion Expression in Modern Literary Appreciation: An Emotion-Based Analysis. *Frontiers in Psychology*, 13.

Churnjeet Mahn. 2016. Travel writing and sexuality : queering the genre. In Churnjeet Mahn, editor, *The Routledge companion to travel writing*, Routledge companions, pages 46–56. Routlege, Taylor & Francis Group, London ; New York.

Deon Mai and Wei Emma Zhang. 2020. Aspect Extraction Using Coreference Resolution and Unsupervised Filtering. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 124–129, Suzhou, China. Association for Computational Linguistics.

Enrique Manjavacas Arevalo and Lauren Fonteyn. 2022. Non-parametric word sense disambiguation for historical languages. In *Proceedings of the 2nd International Workshop on Natural Language Processing for Digital Humanities*, pages 123–134, Taipei, Taiwan. Association for Computational Linguistics.

Stephan Marche. 2012. Literature Is not Data: Against Digital Humanities. Section: culture.

Gunther Martens, Lore De Greve, and Pranaydeep Singh. 2023. A comparison of ChatGPT and fine-tuned BERT with regard to ABSA in the domain of literary criticism - Aspect-Based Sentiment Analysis for Literary Criticism: Experts vs. Social Critics on Literary Prizes.

Claudio Masolo, Emilio Sanfilippo, Marion Lamé, and Perrine Pittet. 2019. Modeling Concept Drift for Historical Research in the Digital Humanities. In *1st International Workshop on Ontologies for Digital Humanities and their Social Analysis (WODHSA)*, Graz, Austria.

Barbara McGillivray, Thierry Poibeau, and Pablo Ruiz Fabo. 2020. Digital Humanities and Natural Language Processing: Je t'aime... Moi non plus. *Digital Humanities Quarterly*, 014(2).

Julie Mennes, Ted Pedersen, and Els Lefever. 2019. Approaching terminological ambiguity in cross-disciplinary communication as a word sense induction task: a pilot study. *Language Resources and Evaluation*, 53(4):889–917.

MIT Global Shakespeare. 2023. Generative AI and the Digital Humanities: Pedagogical Implications.

Franco Moretti. 2014. "Operationalizing": or, the Function of Measurement in Modern Literary Theory. *The Journal of English Language and Literature*, 60(1):3–19.

Mohammad Erfan Mowlaei, Mohammad Saniee Abadeh, and Hamidreza Keshavarz. 2020. Aspect-based sentiment analysis using adaptive aspect-based lexicons. *Expert Systems with Applications*, 148:113234.

Trevor Muñoz and Katie Rawson. 2019. Against cleaning. In *Debates in the Digital Humanities 2019*, pages 279–292. University of Minnesota Press.

M. Devi Sri Nandhini, Pradeep Gurunathan, and Yuvaraj D. 2018. Exploring Aspect-Based Sentiment Analysis – A Survey.

Clemens Neudecker. 2016. An Open Corpus for Named Entity Recognition in Historic Newspapers. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4348–4352, Portorož, Slovenia. European Language Resources Association (ELRA).

Dong Nguyen, Maria Liakata, Simon DeDeo, Jacob Eisenstein, David Mimno, Rebekah Tromble, and Jane Winters. 2020. How We Do Things With Words: Analyzing Text as Social and Cultural Data. *Frontiers in Artificial Intelligence*, 3:62.

Fabian Offert and Peter Bell. 2020. Generative digital humanities. In *Workshop on Computational Humanities Research*.

Michele Pangrazzi. 2022. Building an aspect-based sentiment analysis pipeline using GPT-3.

Aris Pattakos. 2021. Aspect-Based Sentiment Analysis Using Spacy & TextBlob | by Aris Pattakos | Towards Data Science.

Murray G. Phillips and Gary Osmond. 2015. Australia's Women Surfers: History, Methodology and the Digital Humanities. *Australian Historical Studies*, 46(2):285–303.

Luis A. Pineda, Noé Hernández, Iván Torres, Gibrán Fuentes, and Nydia Pineda De Avila. 2020. Practical non-monotonic knowledge-base system for un-regimented domains: A Case-study in digital humanities. *Information Processing & Management*, 57(3):102214.

Barbara Plank. 2016. What to do about non-standard (or non-canonical) language in NLP. *CoRR*, abs/1608.07836.

Barbara Plank. 2022. The 'Problem' of Human Label Variation: On Ground Truth in Data, Modeling and Evaluation. Publisher: arXiv Version Number: 1.

Gabrina Pounds. 2021. The values of trees and woodland: a discourse-based cross-disciplinary perspective on integrating 'revealed' evaluations of nature into environmental agendas. *Critical Discourse Studies*, 18(4):461–480. Publisher: Routledge _eprint: https://doi.org/10.1080/17405904.2020.1752757.

Simone Rebora. 2023. Sentiment Analysis in Literary Studies. A Critical Survey. *Digital Humanities Quarterly*, 017(2).

Sharon Richardson. 2022. Exposing the many biases in machine learning. *Business Information Review*, 39(3):82–89.

Aynat Rubinstein. 2019. Historical corpora meet the digital humanities: the Jerusalem Corpus of Emergent Modern Hebrew. *Language Resources and Evaluation*, 53(4):807–835.

Baumann Ryan. 2015. Automatic evaluation of OCR quality.

Jan Rörden, Doris Gruber, Martin Krickl, and Bernhard Haslhofer. 2020. Identifying Historical Travelogues in Large Text Corpora Using Machine Learning.

Egil Rønningstad, Erik Velldal, and Lilja Øvrelid. 2023. Entity-Level Sentiment Analysis (ELSA): An exploratory task survey. Publisher: arXiv Version Number: 1.

Satvika, Vikas Thada, and Jaswinder Singh. 2021. A Contemporary Ensemble Aspect-based Opinion Mining Approach for Twitter Data. *International Journal of Advanced Computer Science and Applications*, 12(5).

Yves Scherrer, Tanja Samardžić, and Elvira Glaser. 2019. Digitising Swiss German: how to process and study a polycentric spoken language. *Language Resources and Evaluation*, 53(4):735–769.

Thomas Schmidt, M. Burghardt, and Christian Wolff. 2019a. Toward Multimodal Sentiment Analysis of Historic Plays: A Case Study with Text and Audio for Lessing's Emilia Galotti.

Thomas Schmidt and Manuel Burghardt. 2018. An Evaluation of Lexicon-based Sentiment Analysis Techniques for the Plays of Gotthold Ephraim Lessing. pages 139–149, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Thomas Schmidt, Manuel Burghardt, and Christian Wolff. 2019b. Toward multimodal sentiment analysis of historic plays: A case study with text and audio for lessing's emilia galotti. In *Digital Humanities in the Nordic Countries Conference*.

Amina Siba, Rédda Aboura, Réda Kechairi, Mustapha Maatouk, and Bouthaina Sebbah. 2022. Diachronic study (2000-2019) of bioclimate and land use in Tlemcen region, Northwest Algeria. *International Journal of Environmental Studies*, 0(0):1–14.

Thomas Smits and Melvin Wevers. 2023. A multimodal turn in Digital Humanities. Using contrastive machine learning models to explore, enrich, and analyze digital visual historical collections. *Digital Scholarship in the Humanities*, 38(3):1267–1280.

Rachele Sprugnoli. 2017. "Two days we have passed with the ancients...": a Digital Resource of Historical Travel Writings on Italy.

Sprugnoli, Rachele. 2018. *Arretium or Arezzo? A Neural Approach to the Identification of Place Names in Historical Texts*.

Dominik Stammbach, Maria Antoniak, and Elliott Ash. 2022. Heroes, Villains, and Victims, and GPT-3: Automated Extraction of Character Roles Without Training Data. In *Proceedings of the 4th Workshop of Narrative Understanding (WNU2022)*, pages 47–56, Seattle, United States. Association for Computational Linguistics.

Omri Suissa, Avshalom Elmalech, and Maayan Zhitomirsky-Geffet. 2022. Text analysis using deep neural networks in digital humanities and information science. *Journal of the Association for Information Science and Technology*, 73(2):268–287.

Konstantin Todorov and Giovanni Colavizza. 2020. Transfer Learning for Historical Corpora: An Assessment on Post-OCR Correction and Named Entity Recognition. In *CHR*, CEUR Workshop Proceedings, pages 310–339, Amsterdam. Aachen: CEUR-WS.

Anina Troya, Reshmi Gopalakrishna Pillai, Dr. Cristian Rodriguez Rivero, Dr. Zulkuf Genc, Dr. Subhradeep Kayal, and Dogu Araci. 2022. Aspect-Based Sentiment Analysis of Social Media Data With Pre-Trained Language Models. In *Proceedings of the 2021 5th International Conference on Natural Language Processing and Information Retrieval*, NLPIR '21, pages 8–17, New York, NY, USA. Association for Computing Machinery.

Marieke van Erp, Jesse de Does, Katrien Depuydt, Rob Lenders, and Thomas van Goethem. 2018. Slicing and Dicing a Newspaper Corpus for Historical Ecology Research. In Catherine Faron Zucker, Chiara Ghidini, Amedeo Napoli, and Yannick Toussaint, editors, *Knowledge Engineering and Knowledge Management*, volume 11313, pages 470–484. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

Vector Institute. 2023. Aspect-Based Sentiment Analysis Using Prompt Engineering | Vector Applied Intern Project Talks.

Daniela Francesca Virdis. 2023. Ecostylistics: texts, methodologies and approaches. *Journal of World Languages*. Publisher: De Gruyter Mouton.

Miguel Won, Patricia Murrieta-Flores, and Bruno Martins. 2018. Ensemble Named Entity Recognition (NER): Evaluating NER Tools in the Identification of Place Names in Historical Corpora. *Frontiers in Digital Humanities*, 5:2.

Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning Span-Level Interactions for Aspect Sentiment Triplet Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4755–4766, Online. Association for Computational Linguistics.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-Aware Tagging for Aspect Sentiment Triplet Extraction. Publisher: arXiv Version Number: 3.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated neural networks for targeted sentiment analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, pages 3087–3093.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges. ArXiv:2203.01054 [cs].

7. Language Resource References

Language Resources

Bamman, David and Popat, Sejal and Shen, Sheng. 2019. *An annotated dataset of literary entities*. Association for Computational Linguistics. PID https://github.com/dbamman/litbank.

Cambria, Erik and Li, Yang and Xing, Frank Z. and Poria, Soujanya and Kwok, Kenneth. 2020. *SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis*. Association for Computing Machinery, CIKM '20. PID https://sentic.net/.

Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina. 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv. PID https://huggingface.co/google-bert/bert-base-uncased.

Harrison, Chase. 2022. *Langchain*. PID https://python.langchain.com.

Albert Q. Jiang and Alexandre Sablayrolles and Antoine Roux and Arthur Mensch and Blanche

Savary and Chris Bamford and Devendra Singh Chaplot and Diego de las Casas and Emma Bou Hanna and Florian Bressand and Gianna Lengyel and Guillaume Bour and Guillaume Lample and Lélio Renard Lavaud and Lucile Saulnier and Marie-Anne Lachaux and Pierre Stock and Sandeep Subramanian and Sophia Yang and Szymon Antoniak and Teven Le Scao and Théophile Gervet and Thibaut Lavril and Thomas Wang and Timothée Lacroix and William El Sayed. 2024. *Mixtral of Experts*. PID https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics. Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).

Loper, Edward and Bird, Steven. 2002. *NLTK: The Natural Language Toolkit*. arXiv. PID https://www.nltk.org/.

Manjavacas Arevalo, Enrique and Fonteyn, Lauren. 2021. *MacBERTh: Development and Evaluation of a Historically Pre-trained Language Model for English (1450-1950)*. NLP Association of India (NLPAI). PID https://huggingface.co/emanjavacas/MacBERTh.

Ines Montani and Matthew Honnibal and Matthew Honnibal and Adriane Boyd and Sofie Van Landeghem and Henning Peters. 2023. *explosion/spaCy: v3.7.2: Fixes for APIs and requirements*. Zenodo. PID https://spacy.io/.

Rörden, Jan and Gruber, Doris and Krickl, Martin and Haslhofer, Bernhard. 2020. *Identifying Historical Travelogues in Large Text Corpora Using Machine Learning*. arXiv. PID https://github.com/travelogues/travelogues-corpus. ArXiv:2001.01673 [cs].

Early Modern Dutch Comedies and Farces in the Spotlight: Introducing EmDComF and its Emotion Framework

Florian Debaene[x][o], Kornee van der Haven[o], Veronique Hoste[x]
[x]LT^3, Language Technology & Translation Team, [o]Department of Literary Studies
[x]Groot-Brittanniëlaan 45, [o]Blandijnberg 2, 9000 Gent, Belgium
florian.debaene@ugent.be, cornelis.vanderhaven@ugent.be, veronique.hoste@ugent.be

Abstract

As computational drama studies are developing rapidly, the Dutch dramatic tradition is in need of centralisation still before it can benefit from state-of-the-art methodologies. This paper presents and evaluates EmDComF, a historical corpus of 466 both manually curated and automatically digitised early modern Dutch comedies and farces authored between 1650 and 1725, and describes the refinement of a historically motivated annotation framework exploring sentiment and emotions in these two dramatic subgenres. Originating from Lodewijk Meyer's philosophical writings on passions in the dramatic genre (±1670), published in *Naauwkeurig onderwys in de tooneel-poëzy* (Thorough instruction in the Poetics of Drama) by the literary society Nil Volentibus Arduum in 1765, a historical and genre-specific emotion framework is tested and operationalised for annotating emotions in the domain of early modern Dutch comedies and farces. Based on a frequency and cluster analysis of 782 annotated sentences by 2 expert annotators, the initial 38 emotion labels were restructured to a hierarchical label set of the 5 emotions *Hatred*, *Anxiety*, *Sadness*, *Joy* and *Desire*.

Keywords: Early Modern Dutch Theatre, Historical Drama, NLP, OCR, Emotion Analysis

1. Introduction

Drama as a literary genre has been gaining interest in the Natural Language Processing (NLP) research field in recent years. In 2019, the DraCor database (Fischer et al., 2019) established a standardized XML TEI encoding framework, allowing the dramatic tradition in Europe to be described structurally and language-independently. Thanks to digitizing and encoding initiatives of literature throughout Europe in previous decades, the dramatic genre has opened up to computational and comparative research on a European level. Predicting structure in plain text dramas for corpus expansion through encoding enrichment (Pagel et al., 2021), network analysis based on structural drama features (Botond and Bence, 2023; Santa María Fernández and Dabrowska, 2023), coreference resolution (Pagel and Reiter, 2021), emotion analysis (Schmidt et al., 2021a; Dennerlein et al., 2023) and authorial style development in writing tragedies and comedies (Cafiero and Gabay, 2023) have shown how drama is opening up to data-driven analysis and interpretation. In spite of this momentum for European drama, the Dutch dramatic tradition has not yet been object of such structural comparative research. Lacking standardised datasets first and encoding enrichment second, the Dutch dramatic tradition needs centralisation still before it can partake in riding the waves of computational drama analysis.

In this paper, our objective is to propel research on the Dutch dramatic tradition forward by focusing on early modern Dutch comedies and farces, spanning the period from 1650 to 1725. Comedies and farces, traditionally underexposed or deemed inferior in Dutch literary historiography on drama (te Winkel, 1924; Knuvelder, 1964; Erenstein et al., 1996), showcase the importance of desire and imagination in early modern consumption culture by staging characters who experience socially confirming situations or imaginary social expansions in a broad range of economical settings, recognisable to the early modern consumers in the audience (van Stipriaan, 1996; Porteman and Smits-Veldt, 2008). These types of plays, therefore, display cultural social conduct regarding possession and value assignment, revealing the moral, social and emotional dynamics of early modern consumption (Hinnant, 1995; Perry, 2003; Goldstein and Tigner, 2016; Ferket, 2021). We therefore aim to model how desire and its objects are staged in comedies and farces in the Low Countries, and by doing so individuate unexplored patterns in the theatrical representation of the early modern Dutch consumption culture.

First, we relate the creation of the EmDComF corpus, consisting of manually curated and automatically digitised early modern Dutch comedies and farces authored between 1650 and 1725 in txt format, and evaluate the implementation of the Transkribus Print M1 model for automatic corpus expansion (Section 2). Then, we elaborate how we refined a historically motivated emotion annotation framework for early modern Dutch comedies and farces through data-driven clustering algorithms (Section 3). In conclusion, we discuss our findings and discuss future work (Section 4).

OCR	Ground Truth (OCR + GOLD)		MANUAL	
Google Books	Google Books + CENETON	Google Books + DBNL	CENETON	DBNL
217	92	34	108	15

Table 1: Overview of the EmdComF corpus (n=466) and its subsets.

2. The EmDComF Corpus

2.1. Collecting Text Editions

Comedies and farces were productive dramatic subgenres in early modern Dutch society. In total, we collected 466 early modern comedies and farces written by 165 authors in the period from 1650 to 1725.

For the collection of the early modern Dutch comedies and farces, we made use of both open source editions in txt format from the databases Digitale Bibliotheek voor de Nederlandse Letteren (DBNL) and Census Nederlands Toneel (CENETON), and of scanned editions accessible on Google Books using OCR. In our dataset of 466 unique historical plays, we can individuate three subgroups according to their database provenance: there are 123 texts only available in manually curated form, 217 texts are only available in OCR form, and there are two ground truths (GTs) which contain 126 texts for which both manually curated and OCRed texts are available. With the gold-OCR pairs in both GTs, we are able to measure the quality of OCRed texts in the EmDComF corpus in general, as we can compare 126 OCRed texts to their manual references. The distinction between both GTs is maintained throughout this comparison, because their manual text editions correspond differently to OCRed editions due to differing markup implementations. In Table 1, we give an overview of the provenance of the texts collected in the EmdComF corpus.

Two full-text databases provided manually curated texts, roughly making up half of the dataset. DBNL provided 49 and CENETON provided 200 manually curated plays. The other 217 plays were obtained OCRing scans of printed plays made available by Google Books using the Transkribus Print M1 model. This model was chosen to perform OCR, as it is trained on more than 5,000,000 words in 16 languages, among which Dutch, English, French, German, Italian, Spanish and Latin which appear in varying degrees throughout the plays in the EmDComF corpus, from several print typologies, such as the roman and blackletter script, sometimes both used at the same time in the print editions of the plays in the corpus. Finally, Transkribus' Print M1 model digitises text with an acclaimed 2.20% Character Error Rate accuracy according to their website[1], which makes it an interesting model for multilingual text recognition on multiple historical and modern scripts.

We first assess the quality of OCRed texts in the EmDComF corpus, before initiating further downstream content-wise NLP tasks such as sentiment and emotion detection or other profiling analyses. Doing this, we are able to evaluate which aspects of textual information are maintained or lost in the digitisation process using OCR.

2.2. Metrics

We use Character Error Rate ($CER = \frac{S+D+I}{C_{ref}}$) and Word Error Rate ($WER = \frac{S+D+I}{W_{ref}}$) to evaluate the performance of the digitisation process at the text level for gold-OCR pairs in both GTs. CER is the Levenshtein distance (Levenshtein et al., 1966) between predicted characters and their reference characters (C_{ref}), namely the minimal amount of substitutions (S), deletions (D) and insertions (I) needed to transform the OCRed characters into their reference characters, and WER is defined as the Levenshtein distance between predicted words and their reference words (W_{ref}) following the same logic (Neudecker et al., 2021). We report macro-averaged and micro-averaged CER and WER scores, with the former treating CER and WER scores equally regardless of text length and with the latter aggregating error rates cumulatively according to text length.

We complement CER and WER results with vectorisation similarity calculations, considering the averaged cosine similarity of the lexical and semantic vector representation of gold-OCR pairs in each GT on text level, to estimate the textual quality of the digitisation process. Lexical similarity is assessed through three perspectives on the combined vocabularies of the gold-OCR pairs per GT. First, lexical presence is modeled in gold-OCR pairs using a Bag-of-Words (BoW) representation of all gold and OCR word types per GT. Then, token frequency for each word type is captured through a count-based BoW representation. Finally, relative lexical significance is determined using a Term Frequency-Inverse Document Frequency (TF-IDF) representation, where token frequency per word type in each text is weighted based on the combined vocabulary frequency in its gold-OCR collection. Semantic similarity is modeled with a Doc2Vec representation, which considers context and the contextual meaning of words (Řehůřek and Sojka, 2010; Le and Mikolov, 2014).

[1] https://readcoop.eu/transkribus/public-models/

2.3. Cleaning & Preprocessing

Embedded in two separate databases, different markups were used for the original formatting of the manually curated plays. The manual editions from CENETON were automatically extracted from their html hyperlinks and converted to txt format. The manual editions from DBNL were downloaded in txt format. All manually curated editions downloaded in raw txt needed manual and semi-automatic (regex) cleaning to be able to form the ground truth for their respective raw OCR renderings to be compared to, since the manually curated editions incorporate textual noise superfluous and detrimental to this task and since the OCRed texts can only follow the scans of printed editions (Example 1).

1. -(==1==)(»pagina-aanduiding«) DE GEWAANDE
ADVOCAAT, KLUCHTSPÉL.[2]
- DE
GEWAANDE
ADV
CCAAT
KLUCHTSPÉL

We created two GTs, one for each manually curated subset, to get insight into the quality of the OCRed texts. The CENETON GT consists of 92 gold-OCR pairs and the DBNL GT consists of 34 gold-OCR pairs, which means that we OCRed print editions for which manually curated versions are available in the databases. We calculate the CER and WER values for all pairs in the GTs after different preprocessing steps, comparing the OCRed version of a text with its manually curated edition. Finally, we compare the lexical and semantic vector representations of the GTs after each preprocessing step to measure the textual similarity between gold and OCRed texts at text level. Doing this, we can make an informed estimation of the quality of the subset of 217 OCRed texts for which no manually curated text data are available.

Preprocessing steps undertaken to streamline raw gold-OCR pairs as much as possible, are:

1. Removing superfluous tabs and whitespaces.
2. Lowercasing and decoding diacritical marks.
3. Removing punctuation.
4. Automatic word segmentation and spellcheck using Symspellpy's edit distance[3] for out-of-vocabulary (OOV) OCRed tokens, based on the monogram and bigram frequency dictionary of all manually curated data.

Here follows a gold-OCR pairwise comparison per preprocessing step to illustrate the impact of preprocessing on CER and WER scores, where the first example is a gold sentence and the second an OCRed sentence. Hyphens separate the instances.

1) - DE GEWAANDE ADVOCAAT, KLUCHTSPÉL.
- DE GEWAANDE ADV CCAAT KLUCHTSPÉL
$CER: 12.12 \mid WER: 75.0$

2) - de gewaande advocaat, kluchtspel.
- de gewaande adv ccaat kluchtspel
$CER: 12.12 \mid WER: 75.0$

3) - de gewaande advocaat kluchtspel
- de gewaande adv ccaat kluchtspel
$CER: 6.5 \mid WER: 50.0$

4) - de gewaande advocaat kluchtspel
- de gewaande adv caat kluchtspel
$CER: 3.2 \mid WER: 50.0$

2.4. Results

2.4.1. CER and WER

The distribution of micro and macro-averaged CER and WER in the two databases throughout the preprocessing steps in Table 2 is telling for both how the manually curated editions functioned as references in the GTs and how well the OCR performed on the scanned editions.

Micro and macro-averaged CER and WER after preprocessing steps for the CENETON and DBNL GTs show opposite tendencies, with the CENETON GT obtaining better scores for micro-averaging and the DBNL GT obtaining better scores for macro-averaging. This indicates on the one hand that the DBNL GT had better scoring pairs in its collection (n=34) though cumulatively more errors were aggregated, whereas on the other hand CENETON had worse scoring pairs in its collection (n=92) though reaching lower accumulated error rates.

In general, though, the DBNL GT scores better than the CENETON GT, reaching lower CER and WER scores after preprocessing steps. This indicates that the DBNL gold-OCR pairs throughout correspond better textually, so that there are fewer out of gold strings in OCRed text and/or fewer out of OCRed print edition strings in manually curated text. Despite these preprocessing steps, both OCRed and gold texts in the GTs exhibit non-corresponding textual noise, which maintains Character Error Rate (CER) and Word Error Rate (WER) scores, including: OCR mistakes (wrongfully recognised and/or separated characters), structural deviations from the gold texts present in the OCRed texts, such as repeated titles, acts and scenes in headers and footers and (un)succesfull rendered Google Books vignets, and textual deviations from the OCRed texts indicating text structuring elements, occasional manual typos or word segmentation mistakes in the reference.

[2] English: The Presumed Lawyer, Farce.

[3] https://github.com/mammothb/symspellpy

Step	CENETON (n=92)		DBNL (n=34)		COMBINED (n=126)	
Step	CER^{M}	WER^{M}	CER^{M}	WER^{M}	CER^{M}	WER^{M}
1	10.00	17.82	12.59	18.89	10.70	18.11
2	9.29	15.40	7.88	12.00	8.91	14.48
3	8.48	11.60	7.44	10.07	8.20	11.19
4	9.00	9.98	**7.11**	**8.39**	8.29	9.55
Step	CER^{m}	WER^{m}	CER^{m}	WER^{m}	CER^{m}	WER^{m}
1	9.76	17.26	13.13	19.86	10.67	17.96
2	9.09	14.89	8.10	12.36	8.82	14.21
3	**8.36**	11.50	7.65	10.40	**8.17**	11.20
4	8.85	**9.85**	7.31	8.69	8.43	**9.54**

Table 2: Macro-averaged$^{\mathbf{M}}$ and Micro-averaged$^{\mathbf{m}}$ CER and WER scores after preprocessing the CENETON GT, DBNL GT and the combined GTs.

	CENETON (n=92)				DBNL (n=34)			
vectorised	1	2	3	4	1	2	3	4
BoW	77.65	81.93	87.69	**90.12**	74.73	85.81	88.89	**91.50**
COUNT	99.26	99.38	99.31	**99.40**	99.41	99.53	99.52	**99.58**
TF-IDF	96.97	97.39	97.15	**97.91**	97.90	98.31	98.26	**98.73**
Doc2Vec	96.38	98.07	98.81	**98.92**	97.76	99.15	99.46	**99.51**

Table 3: Averaged cosine similarity scores of BoW, count-based, TF-IDF and Doc2Vec vector representations of the CENETON and DBNL GTs at text level after each preprocessing step.

Nonetheless, layout and string normalisation proves to textually align OCR outputs better in both GTs, as shown in Table 2, by removing superfluous tabs and spaces and punctuation and by lowercasing and ignoring accents. In this way, omitted or superfluously inserted accents or punctuation, or non-corresponding upper-cased or lower-cased characters in both gold and OCRed texts do not interfere with CER and WER scores. Using the previous preprocessing steps and a dictionary-based word segmentation and spellcheck algorithm for OOV OCRed tokens in step 4) from the Symspellpy library, we conclude that, despite the inevitable discrepancies caused by non-corresponding textual noise, OCRed text in this dataset on the average of 126 gold-OCR pairs reaches a correspondence of 91.5% on the character level and 90.5% on the word level.

2.4.2. Lexical and Semantic Vectorisation

The CER and WER scores are indicative of the performance of the Transkribus M1 model for digitising scanned print editions of early modern Dutch comedies and farces, and can be further supported by the averaged cosine similarity scores of the lexical and semantic vector representations of the GTs at the text level per preprocessing step to estimate textual quality after the digitisation process.

Comparing the lexical vectorisation of OCRed texts and their reference texts, we measure lexical differences between each text pair by modeling lexical presence on gold and OCR word types (BoW), lexical frequency on token frequency per word type (count-based BoW) and relative lexical significance (TF-IDF) based on the relative token frequency weighted on the gold-OCR combined vocabulary frequency per GT. This way, both the corresponding and deviating vocabulary items from OCRed texts are assessed from 3 lexical perspectives in the lexical similarity calculations. To perform these lexical vectorisations of the gold-OCR pairs, we used scikit-learn (Pedregosa et al., 2011). The semantical vectorisation of OCRed texts and their reference texts is performed by a Doc2Vec model that is trained on each gold-OCR collection per preprocessing step (5 models for CENETON GT, and 5 for DBNL GT), with each a vector size of 300 and a context window of 10 tokens. Gensim was used to obtain semantic Doc2Vec representations of gold-OCR pairs (Řehůřek and Sojka, 2010). The averaged similarity scores of the lexically and semantically vectorised gold-OCR pairs per preprocessing step on text level are found in Table 3.

The averaged cosine similarity scores of the lexical vector representations of the gold-OCR pairs per GT indicate that the BoW vectorisation, which models the vocabulary word types present in text pairs, closely correlates with the reported micro-averaged WER scores for both GTs after preprocessing step 3 and 4. For the DBNL GT after step 3 and 4, BoW scores 88.89 and 91.50 and WER scores 10.40 and 8.69; and for the CENETON GT, BoW scores 87.69 and 90.12 and WER scores 11.50 and 9.85. This correlation is to be expected, since a BoW representation assesses vocabulary presence at the word type level by creating new word types for OOV OCRed tokens and WER quantifies the percentage of these tokens that do not match their reference counterparts. Count-based BoW vector representations modeling the token frequency of vocabulary word types present in text pairs show an almost exact lexical frequency similarity between gold and OCRed texts, indicating that the distribution of terms is highly consistent across both gold-OCR collections, with minimal variation introduced by token frequencies of OCRed word types regardless of the preprocessing steps. TF-IDF vector representations demonstrate a high level of agreement regarding lexical importance in the GTs, yet exhibit a 2.09% deviation in the CENETON GT and a 1.27% deviation in the DBNL GT, underscoring the relative differences in which word types are important based on their token frequencies in correlation to their frequency distribution throughout all gold-OCR pairs per GT. Doc2Vec averaged cosine similarity scores reveal a high and incrementing semantic similarity between gold and OCRed text pairs after each preprocessing step in both GTs. This means that the semantic content and context captured by the Doc2Vec embeddings are becoming increasingly aligned through preprocessing.

Higher similarity scores are, again, generally reported for the DBNL GT, indicating that gold-OCR pairs in the DBNL GT lexically and semantically deviate less than pairs in the CENETON GT, as the former has less and the latter has more non-corresponding textual noise in its OCRed or gold texts on average. Finally, there is a tendency for the similarity scores to increase per preprocessing step in both GTs, with a slight deviation in the count-based and TF-IDF similarity scores for the CENETON GT and DBNL GT after preprocessing step 3 that removes punctuation. Therefore, we find that the proposed preprocessing steps generally lower the distance between the OCRed and gold texts in both GTs by effectively making their lexical and semantic similarities more explicit in all vector representations.

Based on the averaged vectorised comparison of 126 gold-OCR pairs, we conclude that the OCRed texts can be expected to be qualitative enough for further textual analysis despite persisting CER and WER scores averaging around 8.5% and 9.5% respectively due to non-corresponding textual noise, since they capture very similar amounts of lexical and semantic information to their manually curated counterparts. By analogy, this means that the subset of 217 uniquely OCRed texts should convey similar amounts of lexical and semantic information on average after the proposed preprocessing steps, which makes them valuable assets to this dataset. This also suggests the automatic digitisation of early modern Dutch comedies and farces using Transkribus M1 model to be a worthwhile corpus expansion method.

At last, we contend that the presence of textual noise previously identified in both OCRed and gold texts should not necessarily undermine the overall textual quality of the OCRed texts in the EmDComF corpus. In future work, our focus will be on testing the usability of these OCR data through textual analysis. This will best illustrate the real impact of the observed average 2.09% deviation in the relatively significant content words in the OCRed CENETON vocabulary and of the 1.27% deviation in the OCRed DBNL vocabulary on the one hand, and the impact of the semantic deviations of 1.08% for OCRed CENETON texts and 0.49% for OCRed DBNL texts on the other hand. These deviations might eventually be deemed negligible within this corpus, which could have important implications for the automatic corpus expansion of other historical dramatic traditions for which no manually curated but scanned editions are available. Nonetheless, we plan to explore additional OCR post-correction or language normalisation techniques to further process deviations in OCRed texts of the EmDComF corpus.

3. Historical Emotions in EmDComF

The EmDComF corpus consists of 466 early modern Dutch comedies and farces, with OCRed texts that we have demonstrated to display very high lexical and semantic similarities to the manually curated editions on average. Now that the textual quality of both types of text editions has been put into perspective, we proceed to the content-wise emotion analysis of these text data as early modern Dutch comedies and farces have been suggested to particularly display moral, social and emotional dynamics of early modern Dutch consumption culture (Hinnant, 1995; Perry, 2003; Goldstein and Tigner, 2016; Ferket, 2021). After a discussion on Emotion Analysis in historical drama, we describe the refinement of a data-driven genre-specific emotion annotation framework to be implemented in a Machine Learning (ML) approach. With this, we aim to create expert systems capable of automatically detecting emotion within the EmDComF corpus by fine-tuning pre-trained LLMs (large language models) on historical and modern Dutch, such as GysBERT (Manjavacas Arevalo and Fonteyn, 2022) and BERTje (de Vries et al., 2019) respectively, based on the manual annotations of sentiment and emotion in the corpus.

3.1. Emotion Analysis in Historical Drama

Sentiment Analysis (SA) is defined by Liu (2020) as the field of study that analyses people's opinions, sentiments, appraisals, attitudes, and emotions toward entities and their attributes expressed in written text. As a popular application from the

Emotion	A-a	B-a	A-r	B-r	K	F1
Affection	11	13	1.99	2.96	0.66	0.67
Ambition	37	27	6.70	6.15	0.61	0.63
Anger	43	49	7.79	11.16	0.82	0.83
Audacity	4	31	0.72	7.06	0.22	0.23
Aversion	50	53	9.06	12.07	0.76	0.78
Compassion	0	1	0.00	0.23	0.00	0.00
Confidence	81	7	14.67	1.59	0.12	0.14
Consternation	26	15	4.71	3.42	0.48	0.49
Courage	3	2	0.54	0.46	0.40	0.40
Cowardice	0	1	0.00	0.23	0.00	0.00
Curiosity	20	6	3.62	1.37	0.38	0.38
Desperation	8	8	1.45	1.82	0.75	0.75
Devotion	18	8	3.26	1.82	0.61	0.62
Enjoyment	1	6	0.18	1.37	0.28	0.29
Favor	8	9	1.45	2.05	0.94	0.94
Fear	9	9	1.63	2.05	0.66	0.67
Friendship	8	8	1.45	1.82	0.87	0.88
Gratitude	5	6	0.91	1.37	0.54	0.55
Hatred	3	5	0.54	1.14	0.75	0.75
Hope	14	4	2.54	0.91	0.33	0.33
Indecision	1	0	0.18	0.00	0.00	0.00
Indignation	66	41	11.96	9.34	0.61	0.64
Joy	17	9	3.08	2.05	0.45	0.46
Love	23	27	4.17	6.15	0.88	0.88
Peace of mind	2	4	0.36	0.91	0.67	0.67
Pity	1	1	0.18	0.23	1.00	1.00
Pride	17	9	3.08	2.05	0.69	0.69
Regret	12	11	2.17	2.51	0.78	0.78
Remorse	1	1	0.18	0.23	1.00	1.00
Sadness	20	12	3.62	2.73	0.62	0.63
Satisfaction	25	26	4.53	5.92	0.82	0.82
Shame	1	1	0.18	0.23	1.00	1.00
Uneasiness	15	26	2.72	5.92	0.58	0.59
Vindictiveness	2	3	0.36	0.68	0.80	0.80

Table 4: Absolute frequency (a), relative frequency (r), Cohen's Kappa (K), and F1-score for the 34 emotion labels annotated by annotator A and annotator B in the annotation test set of 782 sentences.

NLP domain, SA is nowadays often being used to identify positive, neutral, negative or mixed sentiments expressed in product reviews or social media posts, as well as the targets of these sentiments (Liu, 2020). Emotion Analysis (EA), a subdomain of SA, deals with the more complex task of identifying different emotion classes like joy and sadness in texts, instead of the aforementioned sentiment polarity (Kim and Klinger, 2019; Rebora, 2023). In the last decade, SA and EA have been increasingly applied at the intersection of NLP and Digital Humanities (DH) in the field of Computational Literary Studies, as literary research is often concerned with understanding sentiments and emotions that organise and orient narratives throughout literary genres since the emergence of literary traditions (Hogan, 2011).

In computational literary research adopting SA and EA in historical drama, Leemans et al. (2017) aimed to trace historical changes in emotion expression and in the embodiment of emotions in a corpus of 29 historical Dutch theatre plays from between 1600 to 1800. To this end, the first lexicons and emotion classification schemes for early modern Dutch were created by annotating 27,993 sentences with 38 historically accurate emotion labels, body part labels, bodily process labels, emotional action labels and body sensation labels (van der Zwaan et al., 2015; Leemans et al., 2017). Using a combination of dictionary-based approaches, this first historical emotion classification methodology for early modern Dutch drama reached a 10% precision and 60% recall on the test set.

In historical German drama, state-of-the-art methodologies have been applied for sentiment and emotion classification using transformer-based language models (Schmidt et al., 2021a; Dennerlein et al., 2023). Anchoring their hierarchical emotion annotation scheme in a German literary studies perspective to annotate 13 sub-emotions coming from 6 main emotion classes expressed or attributed to characters (Schmidt et al., 2021b; Dennerlein et al., 2022), Schmidt et al. (2021a) acquired 13,264 annotations from 11 historical German plays and Dennerlein et al. (2023) acquired 11,939 annotations from 17 historical German plays. Both studies evaluated multiple transformer-based ML approaches to classify text sequences with single emotion labels from their emotion framework. Schmidt et al. (2021a) separately report polarity classification accuracy and F1-score up to 90% for their 2 polarity classes (positive/negative), 75% accuracy and F1-score for main emotion class classification and 66% accuracy and F1-score for the 13 sub-emotion classification after fine-tuning on an annotation subset filtered on disagreeing annotations. Dennerlein et al. (2023) report an accuracy of 73% for the 14 sub-emotion classification (a neutral category was added) in cross-validation from which the 6 main emotion classifications and 4 polarity classifications are derived, after fine-tuning on a similarly filtered annotation subset. With this performance, Dennerlein et al. (2023) succeeded in detecting emotional differences between historical German comedies and tragedies.

3.2. Operationalising Historical Emotions

To be able to detect emotions that are historically relevant to the comedies and farces from the EmDComF corpus, we operationalised the historical emotion framework for early modern theatre composed by Lodewijk Meyer around 1670 for emotion annotation. Meyer defined emotions, or passions, as abnormal motions of the heart caused by the notions of good or evil and perceived by the soul (Steenbakkers, 1999). This vision on emotions being caused by individual moralistic judgment about

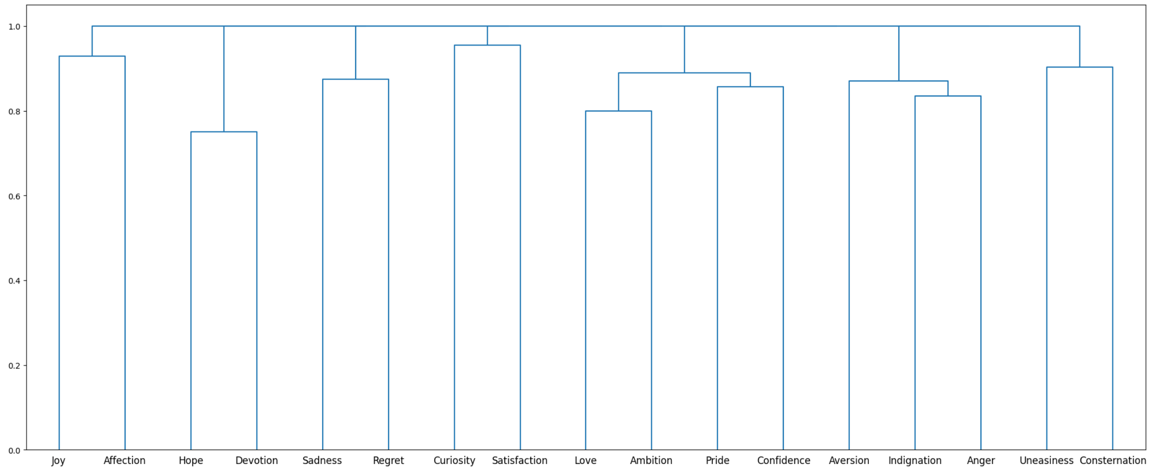


Figure 1: Annotator A's emotion clusters with weighted-linking filtered on infrequent emotions.

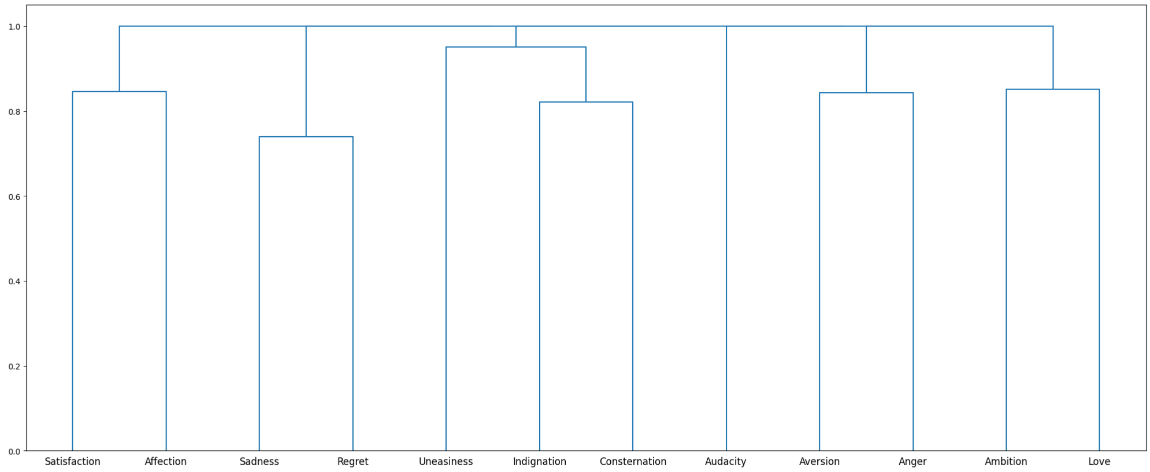


Figure 2: Annotator B's emotion clusters with weighted-linking filtered on infrequent emotions.

good or evil was rooted in contemporary philosophical and literary debates on ethics and human nature. In the instructive work on theatre poetics *Naauwkeurig onderwys in de tooneel-poëzy* (Thorough instruction in the Poetics of Drama) published by literary society Nil Volentibus Arduum in 1765 (Harmsen, 1989; Steenbakkers, 1999), Meyer's moralistic and individualistic conceptualisation of emotion was authoritative in early modern Dutch theatre writing. His description of 38 emotions in the domain-specific context of the EmDComF corpus therefore validates our approach to adopting this emotion annotation framework.

In the annotation study conducted to get insight in the emotionality of the EmDComF corpus, we made use of Meyer's initial 38 emotion labels to annotate emotions expressed or attributed to characters in sentences. Sentiment was annotated on sentence level, using a positive, neutral or negative label. Per sentence, only one sentiment but multiple emotions could be annotated if this was necessary. NLTK Punkt sentence segmentation (Bird et al., 2009) was used to create the sentences, as this greedy sentence splitting method seemed most fit for this task instead of relying on regular expressions. Guided by the description of these 38 emotion categories as summarised by Harmsen (1989) during the annotations, two expert annotators independently annotated emotions and sentiment in one act of a comedy from the corpus, consisting of 782 sentences in authentic early modern Dutch. In these sentences, annotator A annotated 552 emotions, and annotator B annotated 439 emotions using 34 of the 38 emotion labels with varying frequencies. This is due to the fact that delineating emotions in these sentences is at times an interpretative task, which is why the annotators often disagreed in their annotations. In Table 4, we give an overview of the annotation study from the perspective of both annotators per annotated emotion label, the absolute and relative frequencies, and Cohen's kappa (Cohen, 1960) and F1-scores to determine the Inter-Annotator Agreement (IAA).

Throughout the 34 annotated emotion labels, emotion agreement is moderate as the mean Kappa score is 0.59 (0.4 < k < 0.6) and F1-score is 0.60 , whereas sentiment agreement is substantial with a mean Kappa score of 0.75 (0.6 < k < 0.8) and F1-score of 0.85. Nevertheless, class imbalances due to different emotion frequency annotations per annotator were created by this fine-grained annotation set, resulting in a few emotion labels with non-existing or perfect IAA scores. For the emotions *Compassion*, *Cowardice* and *Indecision*, IAA scores are 0.00 as these single-time annotated labels were not used by the other annotator, and IAA for the emotions *Pity*, *Remorse* and *Shame* is theoretically perfect as the single time that these emotions occurred in the annotation set, they were annotated. For example in sentence 656 "Zou hy zich zo verstooren?"[4], annotator A labeled *Indecision* and B labeled *Uneasiness*; in sentence 73 "Ik kon immers 't arme maag're beest zo niet in de open lucht laaten staan."[5], both annotators labeled *Pity*. More frequently occurring emotions like *Ambition*, *Anger*, *Aversion*, *Indignation*, *Love* and *Satisfaction* report more meaningful substantial IAA scores as these were throughout consistently annotated by both annotators. Finally, some emotions like *Audacity*, *Confidence*, *Curiosity*, *Enjoyment* and *Hope* were not annotated with consistent frequency by both annotators, meaning that often the other annotator did not label that sentence or used another emotion label having interpreted it differently. For example in sentence 714 "Zeer wel, daar wil ik wel van snoepen."[6], annotator A labeled *Joy* and B labeled *Enjoyment*.

3.2.1. Clustering Emotion Annotations

Annotating 38 emotion labels remains a challenge as some emotion labels are hard to be distinguished from one another and seem open to interpretation, even though moderate IAA was reached on annotating 34 of the initial 38 emotions found in the annotation test. To operationalise this fine-grained label set and to establish an emotionality framework that fits the EmDComF corpus best, we apply methodological solutions for emotion class imbalances in fine-grained emotional frameworks, illustrated by similar research analysing emotion in another domain. De Bruyne et al. (2019) created a domain-specific emotion set of 5 emotion labels by clustering the annotations from an annotation study labeling 25 emotions in modern Dutch tweets. Their approach increased efficiency in the annotation process by hierarchically structuring the emotion framework, which means that similar emotion labels were grouped together under a broader label that captures a shared emotional essence. Therefore, as hierarchical emotion frameworks have also shown to be effective in research detecting emotion in historical German drama (Schmidt et al., 2021a; Dennerlein et al., 2023), we adopt the clustering methodology to hierarchically structure the 34 annotated emotion labels in the annotations by both annotators as proposed by De Bruyne et al. (2019).

To cluster the annotated emotion labels per annotator, each emotion label is transformed into a vector, resulting in 34 782-dimensional vectors as 782 sentences were annotated. Emotion presence in these vectors is binarised per dimension, with 1 indicating emotion presence and with 0 indicating its absence. These binarised vectors are then fed to hierarchical clustering algorithms performed with SciPy (Virtanen et al., 2020). SciPy's weighted linkage method (WPGMA: $d(u,v) = \frac{\text{dist}(s,v)+\text{dist}(t,v)}{2}$)[7] resulted in the most intuitive dendrograms (emotion label clusters), as it iteratively merges clusters based on the average distances between them, ultimately forming a hierarchical structure that reflects those average linkage distances, and are therefore the only clustering results we report. We applied the weighted linkage method on both annotation sets and on the annotation sets filtered on infrequent emotion labels occurring less than 10 times per set, showcasing the consistent cluster intervals based on the average linkage distances. Figures 1 and 2 show the weighted-linking dendrograms based on both annotators' filtered annotation set.

Concluding, 7 hierarchical clusters result from annotator A's and 5 from annotator B's annotations. Both dendrograms acknowledge main classes for *Joy*, *Sadness*, *Hatred*, *Anxiety* and *Desire*; with annotator A's dendrogram distinguishing another two main classes for *Hope-Devotion* and *Curiosity-Satisfaction*. Based on the emotion frequency and weighted cluster results of the 782 annotated sentences and a historical literary studies perspective on the dramatics and emotions in early modern Dutch comedies and farces, we propose the hierarchical emotion set of 5 labels that we will continue to use in future annotations in Table 5. Based on the dendrograms, we merged *Fear*, *Desperation* and *Cowardice*; *Consternation*, *Uneasiness* and *Indignation*; *Regret* and *Remorse*; *Joy* and *Enjoyment*; *Affection*, *Friendship*, *Compassion* and *Gratitude*; *Satisfaction*, *Peace of mind* and *Relief*; *Devotion* and *Favour*; and finally *Pride*, *Confidence*, *Audacity* and *Courage*. We left the under-represented emotions of *Vindictiveness*, *Indecision*, *Shame* and *Pity* extant to be annotated in future work to de-

[4] English: Would he be so upset?

[5] English: After all, I couldn't leave the poor, skinny animal out in the open like that.

[6] English: Very well, I would like to snack on that.

[7] Weighted Pair Group Method with Arithmetic Mean: $d(u,v)$: distance between clusters u and v. $dist(s,v)$: distance from s to v, $dist(t,v)$: distance from t to v. The formula averages these distances to compute $d(u,v)$.

Label	A-a	B-a	A-r	B-r
Hatred	**98**	**18%**	**110**	**25%**
aversion	50	9%	53	12%
anger	43	8%	49	11%
vindictiveness	2	0%	3	1%
(hatred)	3	1%	5	1%
Anxiety	**126**	**23%**	**101**	**23%**
fear	17	3%	18	4%
indecision	1	0%	0	0%
shame	1	0%	1	0%
consternation	107	19%	82	19%
Sadness	**34**	**6%**	**25**	**6%**
(sadness)	20	4%	12	3%
regret	13	2%	12	3%
pity	1	0%	1	0%
Joy	**69**	**13%**	**73**	**17%**
(joy)	18	3%	15	3%
affection	24	4%	28	6%
satisfaction	27	5%	30	7%
Desire	**225**	**41%**	**130**	**30%**
devotion	26	5%	17	4%
love	23	4%	27	6%
ambition	37	7%	27	6%
pride	105	19%	49	11%
hope	14	3%	4	1%
curiosity	20	4%	6	1%

Table 5: Absolute (a) and relative (r) emotion frequency distribution in the hierarchical label set of 5 emotions and 20 sub-emotions based on the annotations of annotator A and B.

cide if they have their own place in this framework or should be merged. We finally remark that the resulting hierarchical emotion framework of 5 labels seems to correspond quite closely to some modern day emotion classifications, namely with the 5 labels of *Joy*, *Sadness*, *Anger*, *Nervousness* and *Love* established by De Bruyne et al. (2019) or with 4 of the 6 basic emotions linked to universal facial expressions of *Joy*, *Anger*, *Fear* and *Sadness* by Ekman (1992). Other than these emotion frameworks, we explicitly maintain more fine-grained emotion sub-classifications as we hope they will eventually be useful in an Aspect-based Sentiment Analysis (ABSA) methodology to find the objects of the detected emotions in early modern Dutch comedies and farces, with specific regard to the different sub-emotions that were clustered under the label of *Desire*.

4. Conclusion & Future Work

In this paper, we presented and evaluated the EmDComF corpus of 466 early modern Dutch comedies and farces written between 1650 and 1725 in txt format, of both manually curated and OCRed text editions. The quality of OCRed texts in the corpus was measured using CER and WER on 126 gold-OCR text pairs, which resulted in a micro-averaged CER score of 8.43 and a WER score of 9.54 after preprocessing. Finally, we calculated the lexical and semantic vectorisation similarity of 126 gold and OCR texts on text level to further estimate the textual quality of the OCRed texts. These results indicated high lexical and semantic similarities between OCRed texts and their manually curated edition on average, which generally increased after preprocessing. Based on these average CER and WER scores and average lexical and semantic vectorisation similarity scores, we can expect the subset of 217 uniquely OCRed plays in the EmDComF corpus to be similarly qualitative in line with said averages.

Having framed the digitisation performance and the lexical and semantic quality of OCRed plays in the EmDComF corpus, we then related how we refined a historical emotion annotation framework for emotion analysis in early modern Dutch comedies and farces. Lodewijk Meyer's philosophical and literary work on emotions in early modern Dutch theatre provided us the framework of 38 emotion labels. An annotation study on 782 sentences labeling these 38 emotions was conducted by two expert annotators independently. Their annotations indicated a dense emotionality spectrum in the sentences, as 34 emotions had been used in the annotations with a mean Kappa score of 0.59, indicating moderate annotation agreement. Expectedly, annotation sparsity was evident for some emotions using this fine-grained framework. To operationalise the initial emotion framework, clustering algorithms were performed on the annotated emotion labels to establish a hierarchical emotion label set. The refined emotion annotation framework we propose consists of 5 hierarchical emotions *Hatred*, *Anxiety*, *Sadness*, *Joy* and *Desire* with 20 possible sub-emotions based on the clusterisation and the annotation frequencies. Recalculating IAA on the 5 main emotion labels, annotation agreement is now substantial with a mean Kappa score of 0.68 and F1-score of 0.72. *Hatred* has almost perfect agreement with a 0.85 Kappa score; *Joy*, *Sadness* and *Anxiety* have substantial agreement with Kappa scores of 0.79, 0.68 and 0.64 respectively; *Desire* is the hardest category to agree on, having moderate agreement with a 0.43 Kappa score.

Our future plans involve expanding the annotation of plays using this refined emotion framework. These annotations will then serve as training data to fine-tune LLMs, allowing for the creation of expert systems capable of automatically detecting emotion in early modern Dutch comedies and farces. Additionally, we plan to integrate this approach into an ABSA methodology to automatically link identified emotions with their respective objects in the EmDComF corpus, aiming to establish an emotional object typology specific for these types of historical plays with specific regard to expressions of *Desire*.

5. Acknowledgements

This research was carried out with the support of the Research Foundation – Flanders (FWO) under grant G032123N.

6. Bibliographical References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Szemes Botond and Vida Bence. 2023. Tragic and Comical Networks. Clustering Dramatic Genres According to Structural Properties. (arXiv:2302.08258).

Florian Cafiero and Simon Gabay. 2023. Dating the Stylistic Turn: The Strength of the Auctorial Signal in Early Modern French Plays. In *Computational Humanities Research*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Luna De Bruyne, Orphée De Clercq, and Véronique Hoste. 2019. Towards an empirically grounded framework for emotion analysis. In *HUSO 2019: The Fifth International Conference on Human and Social Analytics*, pages 11–16. IARIA, International Academy, Research, and Industry Association.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. (arXiv:1912.09582).

Katrin Dennerlein, Thomas Schmidt, and Christian Wolff. 2022. Figurenemotionen in deutschsprachigen Dramen annotieren. Zenodo.

Katrin Dennerlein, Thomas Schmidt, and Christian Wolff. 2023. Computational emotion classification for genre corpora of German tragedies and comedies from 17th to early 19th century. *Digital Scholarship in the Humanities*, 38(4):1466–1481.

Paul Ekman. 1992. An argument for basic emotions. volume 6, pages 169–200, United Kingdom. Taylor & Francis.

Robert L. Erenstein, Dirk Coigneau, Robert van Gaal, Flor Peeters, Herman Pleij, Karel Porteman, Jaak van Schoor, and Mieke B. Smits-Veldt. 1996. *Een Theatergeschiedenis Der Nederlanden. Tien Eeuwen Drama En Theater in Nederland En Vlaanderen*. Amsterdam University Press.

Johanna Ferket. 2021. *Hekelen Met Humor: Maatschappijkritiek in Het Zeventiende-Eeuwse Komische Toneel in de Nederlanden*. Uitgeverij Verloren.

Frank Fischer, Ingo Börner, Mathias Göbel, Angelika Hechtl, Christopher Kittel, Carsten Milling, and Peer Trilcke. 2019. Programmable Corpora: Introducing DraCor, an Infrastructure for the Research on European Drama. Zenodo.

David B. Goldstein and Amy L. Tigner. 2016. *Culinary Shakespeare: Staging Food and Drink in Early Modern England*. Penn State Press.

Antonius Johannes Engbert Harmsen. 1989. *Onderwys in de tooneel-poëzy: de opvattingen over toneel van het Kunstgenootschap Nil Volentibus Arduum*. Ordeman.

Charles H. Hinnant. 1995. Pleasure and the Political Economy of Consumption in Restoration Comedy. *Restoration: Studies in English Literary Culture, 1660-1700*, 19(2):77–87.

Patrick Colm Hogan. 2011. *Affective Narratology: The Emotional Structure of Stories*. University of Nebraska Press.

Evgeny Kim and Roman Klinger. 2019. A Survey on Sentiment and Emotion Analysis for Computational Literary Studies. *Zeitschrift für digitale Geisteswissenschaften*.

Gerard Petrus Maria Knuvelder. 1964. *Handboek Tot de Geschiedenis Der Nederlandse Letterkunde*. Malmberg, Den Bosch.

Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. (arXiv:1405.4053).

Inger Leemans, Janneke M. van der Zwaan, Isa Maks, Erika Kuijpers, and Kristine Steenbergh. 2017. Mining Embodied Emotions: A Comparative Analysis of Sentiment and Emotion in Dutch Texts, 1600-1800. *Digital Humanities Quarterly*, 11(4).

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Bing Liu. 2020. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, 2 edition. Studies in Natural Language Processing. Cambridge University Press, Cambridge.

Enrique Manjavacas Arevalo and Lauren Fonteyn. 2022. Non-parametric word sense disambiguation for historical languages. In *Proceedings of* *the 2nd International Workshop on Natural Language Processing for Digital Humanities*, pages 123–134, Taipei, Taiwan. Association for Computational Linguistics.

Clemens Neudecker, Konstantin Baierer, Mike Gerber, Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. 2021. A survey of ocr evaluation tools and metrics. In *Proceedings of the 6th International Workshop on Historical Document Imaging and Processing*, HIP '21, page 13–18, New York, NY, USA. Association for Computing Machinery.

Janis Pagel and Nils Reiter. 2021. DramaCoref: A Hybrid Coreference Resolution System for German Theater Plays. In *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 36–46, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Janis Pagel, Nidhi Sihag, and Nils Reiter. 2021. Predicting Structural Elements in German Drama. In *Proceedings of the Second Conference on Computational Humanities Research*, volume 1613, page 0073.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Curtis Perry. 2003. Commerce, Community, and Nostalgia in The Comedy of Errors. In Linda Woodbridge, editor, *Money and the Age of Shakespeare*, pages 39–51. Palgrave Macmillan US, New York.

K. Porteman and Mieke B. Smits-Veldt. 2008. *Een nieuw vaderland voor de muzen: geschiedenis van de Nederlandse literatuur, 1560-1700*. Bakker.

Simone Rebora. 2023. Sentiment Analysis in Literary Studies. A Critical Survey. *Digital Humanities Quarterly*, 017(2).

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Luis Rei and Dunja Mladenić. 2023. Detecting Fine-Grained Emotions in Literature. volume 13, page 7502. Multidisciplinary Digital Publishing Institute.

María Teresa Santa María Fernández and Monika Dabrowska. 2023. Análisis comparativo del Coro como personaje en tres tragedias griega y tres dramas españoles del Corpus DraCor. *Neophilologus*, 107(3):389–412.

Thomas Schmidt, Katrin Dennerlein, and Christian Wolff. 2021a. Emotion Classification in German Plays with Transformer-based Language Models Pretrained on Historical and Contemporary Language. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 67–79, Punta Cana, Dominican Republic (online). Association for Computational Linguistics.

Thomas Schmidt, Katrin Dennerlein, and Christian Wolff. 2021b. Towards a Corpus of Historical German Plays with Emotion Annotations. In *3rd Conference on Language, Data and Knowledge (LDK 2021)*, volume 93 of *Open Access Series in Informatics (OASIcs)*, pages 9:1–9:11, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Piet Steenbakkers. 1999. The Passions according to Lodewijk Meyer: Between Descartes and Spinoza. In *Desire and Affect: Spinoza as Psychologist ; Papers Presented at the Third Jerusalem Conference (Ethica III)*, pages 193–210.

J. te Winkel. 1924. *De Ontwikkelingsgang Der Nederlandsche Letterkunde, Deel IV. Geschiedenis der Nederlandsche letterkunde van de Republiek der Vereenigde Nederlanden (2)*. De erven F. Bohn, Haarlem.

Janneke M. van der Zwaan, Inger Leemans, Erika Kuijpers, and Isa Maks. 2015. HEEM, a complex model for mining emotions in historical text. In *2015 IEEE 11th International Conference on E-Science*, pages 22–30. IEEE.

René van Stipriaan. 1996. *Leugens en vermaak: Boccaccio's novellen in de kluchtcultuur van de Nederlandse renaissance*. Amsterdam University Press.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

When Hieroglyphs Meet Technology: A Linguistic Journey through Ancient Egypt Using Natural Language Processing

Ricardo Muñoz Sánchez
Språkbanken Text, University of Gothenburg
ricardo.munoz.sanchez@svenska.gu.se

Abstract

Knowing our past can help us better understand our future. The explosive development of NLP in these past few decades has allowed us to study ancient languages and cultures in ways that we couldn't have done in the past. However, not all languages have received the same level of attention. Despite its popularity in pop culture, the languages spoken in Ancient Egypt have been somewhat overlooked in terms of NLP research. In this survey paper we give an overview of how NLP has been used to study different variations of the Ancient Egyptian languages. This not only includes Old, Middle, and Late Egyptian but also Demotic and Coptic. We begin by giving a short introduction to these languages and their writing systems, before talking about the corpora and lexical resources that are available digitally. We then show the different NLP tasks that have been tackled for different variations of Ancient Egyptian, as well as the approaches that have been used. We hope that our work can stoke interest in the study of these languages within the NLP community.

Keywords: Ancient Egypt, Ancient Languages, Coptic, Demotic, Historic Languages, Literature Review, Low-Resource Languages

1. Introduction

Ancient Egyptian culture has been called one of the cradles of western civilization (Maisels, 1998). However, there is still much that we do not know about it. The Egyptian people left behind vast amounts of primary textual sources, which the dry weather of the desert helped preserve even if it was in a fragmentary manner. As an example of this, we can take the Oxyrhynchus papyri, a collection of over 500,000 papyri containing fragments of texts, currently housed at the University of Oxford.[1] All of these documents can give us invaluable insights into the lifestyles that these people led and the state of the world at that time. It also can provide unique insights into how technology, science and religion have evolved over time. Developing computational approaches can help us better understand the languages within these documents and how they connect to their environment, while helping preserve them for future generations.

Some issues are quick to appear when attempting to use NLP for Ancient Egyptian. First and foremost is that there are no longer any native speakers left. This means that we cannot know how the language was pronounced[2] or clarify any doubts we may have about the documents. As for making linguistic annotations and translations, it will often take much longer than for living languages (Polis et al., 2015). Furthermore, some of the subtleties of the text might be missed due to lack of the relevant sociocultural context.

Another major issue is that the Ancient Egyptian language was used for over 3,000 years. The vast expanse of the Ancient Egyptian empire and the lack of quick and inexpensive media of transportation lead to major variations in the language (Bard, 2005). More details on the language and on these variations will be discussed in section 2.

Finally, even though a lot of documents survived, most of them are at least partly damaged due to weather conditions, human intervention or just the passage of time. This means that, even if we can extract the whole meaning of the sentence, some nuances or regional variations can be lost to history.

All of these issues mean that the different variations of Ancient Egyptian are considered low-resource languages (Zeldes and Schroeder, 2016; Nederhof and Rahman, 2015). This means that most of the cutting-edge strategies such as transformers (Vaswani et al., 2017) cannot be used for these languages, as those often require vast amounts of data.[3]

For this literature review, we made a survey of the Natural Language Processing (NLP) techniques that have been used recently to study the Ancient Egyptian language. This includes not only the actual implementations, but also some of the difficulties they faced, how they were able to overcome them and some of the implications of their works.

[1] https://www.ees.ac.uk/papyri

[2] Despite this, at least one paper has attempted to do automated pronunciation mining for several dead languages, including Ancient Egyptian and Coptic (Lee et al., 2020).

[3] It should be noted that this is not necessarily the case for Demotic, as it has parallel corpora that allow it to be used for multilingual approaches, see Choudhary and O'riordan (2023) or Khakhmovich et al. (2020) for examples of this.

LT4HALA 2024@LREC-COLING-2024, pages 156–169
25 May, 2024.

We looked for papers dealing with computational approaches and Ancient Egyptian in the ACL Anthology[4], the ACM Digital Library[5], and Google Scholar[6]. We then filtered the papers that are related to NLP. More specifically, we decided not to talk about optical character recognition or the digital representation of the characters either, as we consider those to be image recognition and data representation tasks, respectively, as opposed to NLP ones. We have included all works that match our criteria until January 2024.

It is important to note that we focused mainly on Middle and Late Egyptian as barely any NLP work has focused on Old Egyptian and Demotic. We also focus on Coptic, as this language can also be considered a variation of Ancient Egyptian (Bard, 2005) and a good amount of work has been done for it.

As for the organization of the rest of this paper, we first describe the language in Section 2 and make some comments about it in order to showcase common issues that arise when working with the language. We talk about the corpora available in Section 3, including the kinds of annotations they have and the periods over which they have been updated. In Section 4 we talk about the NLP tasks that are relevant for Ancient Egyptian. Finally, we devote Section 5 to the current state of the use of NLP techniques for Coptic. Even though it still can be considered an evolution of the Ancient Egyptian language, it has a completely different writing system and we have a greater amount of well-preserved documents. As a result, the issues faced when dealing with Coptic are different than those that we face with Ancient Egyptian.

2. The Language

Nederhof and Rahman (2015) provide a good overview of the Ancient Egyptian language and its characteristics in their paper. It is the main source of the information in this section, along with the introduction to hieroglyphs given by Kamrin (2004) and the description of the language given by Bard (2005). However, most of the papers that we mention throughout this literature review also have a brief explanation of the language.

Ancient Egyptian is a language in the Afro-Asiatic family. This family includes the Semitic languages (Hebrew, Arabic, etc.). In the languages of this family the vowels are usually not written and Ancient Egyptian is no different[7]. This, coupled with the fact that there are no native speakers alive, means that we cannot really know how Ancient Egyptian sounded like. Some of the approximations we currently have are made taking into account how phonetics work in the other languages of the family, but we should not fall into the trap of considering them how the language actually sounded.

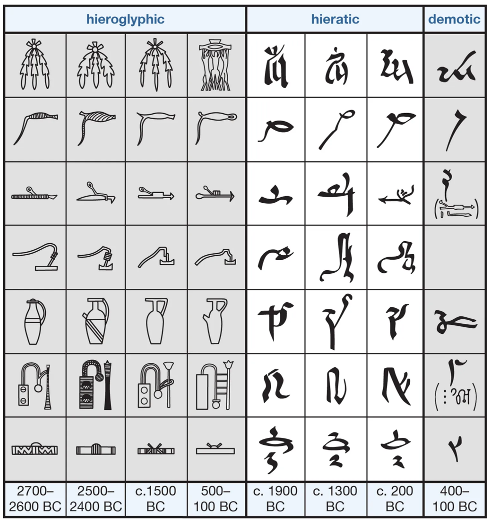


Figure 1: A table illustrating how the Ancient Egyptian scripts evolved over time. It compares seven symbols in hieroglyphic, hieratic, and demotic scripts. Taken from the Encyclopaedia Britannica website,[9] based on the same table by Möller (1919, p. 78).

The writing system was hieroglyphic, but it could also be written in hieratic, a manuscript version of hieroglyphs. An example of how these writing systems evolved over time can be seen in Figure 1. We have included more examples of how these script systems look like in Appendix A. The symbols of this writing system can be divided into logographs, phonographs, determinatives or typographical signs.

Logographs represent either whole words or ideas. That means that a single symbol can represent a complete idea, such as a river or a bird. Phonographs, on the other hand, represent sounds. Each phonograph can correspond from one to three consonants, depending on the symbol. Determinatives help clarify the meaning of the word or disambiguate between otherwise identically written words. Finally, typographical signs are used to give semantic meaning to the word or as fillers.

There are some important considerations that must be taken into account when trying to parse these symbols. Some words can be written either using logograms, just phonograms or a combina-

[4] https://aclanthology.org/

[5] https://dl.acm.org/

[6] https://scholar.google.com/

[7] With the exception of Coptic, where vowels are written.

[9] https://commons.wikimedia.org/wiki/File:Leaves_from_a_Coptic_Manuscript_MET_sf21-148-1as3.jpg (Accessed March 30, 2024)

tion of the two (like in Japanese). Also, some symbols can have more than one function and there are neither end-of-word nor end-of-sentence markers. Furthermore, scribes took into account the aesthetic value of their work, adding or removing symbols as they deemed appropriate. Along the same vein, while the language was written from top to bottom, it could be written from left to right or from right to left and the orientation of the text could be either vertical or horizontal. This means that there is no standardized way of writing the language.

The language also had important variations throughout its history. The Ancient Egypt empire lasted for around 3,000 years and is usually divided into the Old, Middle and New Kingdoms. Between these kingdoms there were periods of great unrest, which lead to big cultural changes. Because of that, the Ancient Egyptian language can be divided into these same stages, with Old and Middle Egyptian being sometimes grouped into Classical Egyptian due to their similarity. However, Late Egyptian does show important differences when compared to Middle Egyptian, both grammatical and morphological, and is often considered as a different language.

Finally, Demotic and Coptic can also be considered later stages of Ancient Egyptian, even though they do not use neither hieroglyphs nor the hieratic script any longer (Bard, 2005). They can also have bigger variations in terms of morphological and grammatical variation, as evidenced by the greater amount of usage of suffixes and the lack of repetition of phonemes in Coptic (Zeldes and Schroeder, 2016).

It is because of all these reasons that most papers just focus on one of the stages of the language instead of trying to focus on all of its history at the same time.

3. Corpora and Lexical Resources

An important first step in order to do any kind of NLP is to have corpora available. However, when studying ancient languages we have the major issue that there are no longer any native speakers to annotate sentences or documents. This in turn means that it takes much longer for them to be annotated (Polis et al., 2015). Here we present the most recent and most comprehensive corpora for the different stages of Ancient Egyptian that we mentioned in Section 2.

3.1. Middle Egyptian

While there were attempts at making corpora of annotated Middle Egyptian, it was until 2017 when Nederhof and Rahman (2015) annotated a corpus for hieratic transliteration that also included the function of each symbol. Taking into consideration that the current NLP approaches do not use the spatial relations of the script, they linearized the text. They also removed variations of symbols, considering that they would do more harm than to help training the models. The corpus currently consists of only two texts. Due to how some words tend to be often repeated throughout each text, its creators suggest to train it on one of them and test it in the other. They argue that, even though mixing both texts allows for more training data, doing so would skew the results of machine learning models and give a false sense of confidence due to data leakage. The corpus is available as part of the larger St. Andrews corpora.[10]

3.2. Late Egyptian

The Ramses project is the most ambitious project regarding Ancient Egyptian corpora, as it is an attempt to build a comprehensive annotated corpus of all available texts in Late Egyptian (c. 1350-700 BC). The project began in 2008, and a first version of their software was first made publicly available in 2013 by Polis et al. (2013). A beta of an online version was released in 2015 (Polis et al., 2015). At the time of its presentation, the corpus had already more than 1350 texts, which amount to over a million words. When the website was announced, it already had over 4000 texts and, during a presentation in 2017 (Polis and Razanajao, 2017), it was announced that the corpus was nearing 5000 texts.

An important feature of this corpus is that from its inception, it included the documents that are considered the most useful for studying the language, along with other texts considered to be relevant for linguistic analysis. The corpus's annotations focus heavily on inflections, lemmata, and spellings, but also include all of the relevant metadata for each text, along with annotations on the state of preservation of the documents (or sections of them) and on alterations or editings of the texts. It also allows the annotators to include comments or criticism on their choices, with references that justify them. Their original paper (Polis et al., 2013) also includes a small tutorial on how to use their software and a list of ways to further expand the project, one of which was including syntactic analysis of the texts.

The online version is currently available at the project website.[11] However, this is only the beta version of the website, which is only available in French and provides access to only a small portion of the corpus. Another issue is that the last update to the website was made in 2016, though Polis and Razanajao (2017) noted in 2017 that the project

[10] https://mjn.host.cs.st-andrews.ac.uk/egyptian/texts/

[11] http://ramses.ulg.ac.be/

was still alive.

3.3. Demotic

The Chicago Demotic Dictionary (Johnson, 2001) is one of the few lexica available for Demotic. It was maintained and updated from 1972 to 2012 and includes not only the words themselves, but also scans of the actual documents. The 2002 edition can be found on the project's website as a PDF document.[12]

3.4. Coptic

A comprehensive corpus of Coptic was created in 2013 and released in 2016. This corpus, called the Coptic Scriptorium (Schroeder and Zeldes, 2016), was designed to be used to study a wide variety of subjects, from linguistics to biblical studies, and consists of eleven smaller corpora. At the time of its release, it had a little less than 60 thousand manually annotated words. This corpus can be used for a wide variety of NLP tasks, most of which can be consulted at the project's website.[13] Most notably, it covers a wide variety of annotations, from tokenization (i.e. identifying the words in a document) all the way to parts-of-speech tagging and a treebank which follows the universal dependencies notation. This is an ongoing project that currently has around 850 thousand annotated words and the documents have enough metadata to tell whether these annotations were made automatically or whether they were either made or revised by humans. Their most recent release was on October 2023 and the current status of the project can be found at their blog.[14]

Several other lexicons for Coptic have been created through time. There is also the Database and Dictionary of Greek Loanwords in Coptic[15], which contain Coptic Lemmas that were adopted from Ancient Greek lemmas. The Marcion project[16] is another lexicon freely available online, with over 11 thousand head words and over 87 thousand items. Both of these lexicons were based on an already existing dictionary (Crum, 1939).

In return, both of these lexicons along with the Coptic section of the TLA were used to create both an online dictionary (Feder et al., 2018) and WordNet (Slaughter et al., 2019). Both of these have been incorporated into the Coptic Scriptorium and its other resources.

Some multilingual collections of corpora contain data in some of the variations of Ancient Egyptian. The Coptic Scriptorium corpus mentioned previously forms part of the Universal Dependencies framework (Zeldes and Abrams, 2018; de Marneffe et al., 2021), a project whose aim is to create a framework for consistent grammatical annotations across different languages. Finally, the OPUS corpora (Tiedemann, 2016) contains parallel data for translation, one of the languages included being Coptic.

3.5. Various Time Periods

The Thesaurus Linguae Aegyptiae (TLA) (Seidlmayer, 2011) was a corpus released in 2004 and was updated until 2012. It contains a wide variety of texts, ranging all the way from the Old Kingdom to the Roman times, including the oldest pyramid texts. This amounts to almost a million and a half words, containing texts in Old, Middle and Late Egyptian, Demotic, and Coptic. It is one of the few annotated Old Egyptian and Demotic corpora. The corpus only has lemmatization and morpho-syntactic annotation and most of their website, including the handbook on how to access and use the database, is in German. The corpus is freely available online.[17]

The Thot Sign List (TSL) (Polis et al., 2021) is a collection of graphemes that have been attested in hieroglyphic or hieratic texts. Its first release contains 1,203 signs, 4,842 functions, and 21,834 tokens. The TSL is freely available on the project website,[18] but a (free) account is necessary to access all of its features.

Nordhoff and Krämer (2022) created a dataset with morpheme annotation for several low-resource languages. It contains examples in Old and Late Egyptian, as well as in Coptic. However, they do not mention the corpus size for any of the languages included.

4. NLP for Middle and Late Egyptian

Rosmorduc (2015) gives a quick overview of some of the main tasks that have been tackled from the 90s to 2015. He notes that, other than some attempts in the 90s, most of the work up until recently had been geared towards creating a standard Unicode representation of hieroglyphs. The most recent updates in this regard were in 2019 and 2021 (Nederhof et al., 2019; Glass et al., 2021), when some control characters to signal some spatial properties of the characters were introduced.

[12] https://oi.uchicago.edu/research/projects/chicago-demotic-dictionary-cdd-0

[13] https://copticscriptorium.org/tools

[14] https://blog.copticscriptorium.org/

[15] https://www.geschkult.fu-berlin.de/en/e/ddglc/index.html

[16] http://marcion.sourceforge.net/dictionary/coptic.html

[17] http://aaew.bbaw.de/tla/

[18] http://thotsignlist.org

4.1. Transliteration

We currently have a very good understanding of how Ancient Egyptian script works, even going as far as having developed standardized methods of transliteration to Latin script and designed Unicode symbols for hieroglyphic script (Nederhof et al., 2019). However, most of these methods require human annotators to work on the text due to the lack of standardization in how the language was written (see section 2). This means that transliteration is still an open problem in the Ancient Egyptian machine learning field.

As mentioned in Section 3, an important issue is that annotation of Ancient Egyptian is a slow process. Because of this, any major breakthrough would mean that more manpower would be available for other tasks in Egyptology.

One of the latest approaches for transliteration is the one by Nederhof and Rahman (2017). They made a probabilistic automaton that can transliterate a text in Middle Egyptian hieratic (i.e. manuscript hieroglyphs) to its phonetic values. For this, they created the Middle Egyptian corpus mentioned in Section 3. It has annotations for the functions of each symbol so as to help the model learn. They consider that the innovation of their system is that it does more than just doing a simple transliteration, it also makes notes on semantic elements of the text. Due to the scarcity of annotated texts from that era, they compare n-gram models (with n varying from 1 to 3) and Hidden Markov Models (HMM). They were able to reach recall and precision scores of approximately 0.95 when interpolating the results from the 3-gram and HMM models. The authors mention that, even though the model used was very basic, this is an important stepping stone for transliterating documents from this era.

In a previous work, Nederhof (2009) notes that alignment could be another possible way to approach transliteration. The proposed model assumes that the signs in the text can only be either phonograms or determinatives, thus ignoring logographs and typographical signs. Moreover, it also assumes that the text can be read without skipping signs or repeating phonograms. In order to make the model more robust, it assigns a penalty to words that could break these rules. The word boundaries are then chosen as the configuration that minimizes this penalty through the use of beam search. When using a simpler text he got an accuracy of 0.98 while experimenting with variations of the model, while a more complicated text got an accuracy of 0.97. He does note, however, that the model might struggle with unseen and/or more complex texts due to things such as unusual ways that words might be written.

Rosmorduc (2009) tried another approach to transliteration. He derived a set of rules on how words are formed and created a series of transducers, that is, finite-state automatons that parse the words and use these rules to verify whether a word is valid or not. The validation set was one of the same texts that Nederhof and Rahman (2015, 2017) used for their corpus and his model achieved a precision of around 0.91. However, this was the same set from which the rules were derived. When using another text as a test set, the precision dropped to 0.82. He justifies his results by claiming that they were due to some small technical errors. Finally, he tried to use the same model on a Late Egyptian text. Even though the precision score for this test is not reported and the author notes that it is quite bad, he mentions that it is on par with what he would expect for a student that has only studied Middle Egyptian but not any of its latter variants.

A later paper by Barthélemy and Rosmorduc (2011) compares two kinds of transducers, but does not report performance scores for either of the models.

Similarly, Bédi et al. (2022) present a multimodal system for transcribing or transliterating endangered and extinct languages (depending on whether the modality is audio or text, respectively). They tested their model on Ancient Egyptian inscriptions, but do not report any quantitative results. A later paper shows how this system would work with a sample text (Bédi et al., 2022), which is also available online.[19]

Finally, Wiesenbach and Riezler (2019) use transcription and part-of-speech tagging as an intermediate step towards translation into German. They used encoders and decoders to achieve these joint tasks. Given that they do not report results for the transliteration, we will talk about their approach in the following section.

4.2. Translation and Part-of-Speech Tagging

Even though translation and part-of-speech (POS) tagging are completely separate tasks, the only paper (to the best of our knowledge) that tackles these tasks in Ancient Egyptian does it in tandem. It should be noted that only the results for the translation task are reported.

Wiesenbach and Riezler (2019) compare different approaches for translating Middle Egyptian into German. These model several tasks jointly under the assumption that it would help with the small amount of data available. They compare using hieroglyphs and their transcription for translation (the many-to-one approach); using hieroglyphs to translate, transcribe, and extract POS tags at the

[19] https://c-lara.unisa.edu.au/lara_legacy/hieroglyphics1avocabpages/_hyperlinked_text_.html

same time (the one-to-many approach); and using both hieroglyphs and their transcription to translate, transcribe, and extract the POS tags (the many-to-many approach). As a baseline with which to compare these approaches they use a system that directly translates hieroglyphs to German.

Their models have an encoder for each type of input and a decoder for each type of output (depending on the approach). These are based on a GRU[20] architecture with attention. They experimented both with a more shallow network of one layer and a deeper one of four layers. For the learning process they compare different schedules to determine whether to lend more weight to the main task (translation) or to the assistance tasks. The data they used was a subset of the Thesaurus Linguae Aegyptiae (TLA) (Seidlmayer, 2011) mentioned in Section 3.

The best performance of their baseline system is a BLEU score of 19.86 points. This score is improved for the best many-to-one system to 21.61 points and to 22.79 points for the best one-to-many system. Meanwhile, the many-to-many system showed no improvement over the baseline, with a BLEU score of 18.07. Thus they conclude that jointly translating, transliterating, and doing POS tagging yields better results than doing a direct translation. It is of note that they do not report results neither on the transcription task nor on the POS tagging task.

4.3. Text Classification

Automatic text classification is another important task in NLP, as it can help document organization and management, text filtering or sense disambiguation. This is particularly useful for ancient languages as it allows us to study them without having to sift through and manipulate the original documents.

Gohy et al. (2013) mention that doing text classification can also give us insights into the registers used for different kinds of texts, which in turn should help improve the performance of machine learning techniques in other NLP tasks. They further claim that this is an important endeavor in the case of dead languages such as Late Egyptian.

In their paper Gohy et al. (2013) did genre classification. The genres they chose were letters, judicial documents, oracular questions, educational texts, monumental inscriptions, hymns and administrative texts. The authors argue that, while assuming that different genres do not overlap is an oversimplification, when chosen carefully they should be relatively independent from each other. They also note that another strong assumption that they are making in their paper is that each genre will have one and only one register and that each register will be exclusive to one genre, which is not true in general. Finally, as they are only interested in the registers, their models use mainly just semantic and morpho-syntactic features, while mostly ignoring the metadata and the structure of the texts.

The models that they used were a naïve Bayes classifier, an SVM, and a segment and combine method (which learns from each syntactic property of the document and then combines what it learnt to get further insights). Their best performing model was the naïve Bayes classifier, which achieves a recall of slightly over 0.84 in general and of over 0.97 with both letters and monumental inscriptions. They consider that in the case of the monumental inscriptions this is due to the more rigid structure used for the language and in the case of the letters it is due to the higher volume of training data available. On the other hand, this model gets a recall of only 0.66 with oracular texts. The authors consider that this is because oracular questions were usually very short (usually one or two sentences) and dealt with daily life matters thus being mostly misclassified as letters. Therefore, they created a modified naïve Bayes classifier which takes into account the length of the texts. This new model improved the recall of oracular questions to over 0.9 and got a general recall improvement of approximately 3%. Their SVM model got similar, but slightly worse results, while the segment and combine model got much more extreme results, with letters, judicial and educational documents, and monumental inscriptions getting a recall of over 0.9, but oracular questions and administrative texts having a recall lower than 0.3.

4.4. Text Retrieval

One of the NLP tasks that would be the most useful for egyptologsts is text retrieval. This task allows to create systems capable of searching and querying indexed documents. Using these kinds of systems would save researchers the effort of sifting through piles of useless data. They also function as a cultural preservation tool, by diminishing the amount of manipulation suffered by the actual physical documents.

In their paper, Iglesias-Franjo and Vilares (2020) created a text information retrieval system for Middle Egyptian. They consulted several egyptologists in order to determine the needs of such a system, most of which were either simplicity of use, flexibility and adhering to the current standard practices of the field. The system first preprocesses and normalizes the text of the documents. The normalization step refers to the way the hieroglyphs are tokenized into "sign groups" as opposed to each symbol being taken separately. After this, an index

[20] GRU stands for gated recurrent unit, a kind of recurrent neural network (Cho et al., 2014).

is created and stored. Once the index is in place, queries can be made. These can be made in latin script, hieroglyphs or a combination of the two. The text is then normalized as in the indexing stage, with the difference that a query using hieroglyphs can specify whether the symbols are the only ones appearing or if the user is looking for words that contain those symbols. Then, a list is selected and ranked according to a Boolean model and a vector space representation of the documents. The authors note that this is a first release and that there is still much work to be done. The system is freely available at their GitHub page.[21] Another approach that they proposed was using a method similar to those used for Japanese dictionaries, where words can be searched by using a combination of kanji (ideograms) and kana (syllabary). However, this query method was considered too unintuitive by the authors. They also note that completion of the Ramses or the Thesaurus Linguae Aegyptiae corpora mentioned in Section 3 could be a great boon to these kinds of systems.

4.5. Semantic Representations

Even though Ancient Egyptian lacks the amount of text needed to create embeddings (either contextual or non-contextual), that does not mean that useful semantic representations cannot be made.

Semantic maps (Georgakopoulos and Polis, 2018) are graphs of meanings such that two meanings are connected to each other if there is a language in which the same linguistic item is used for both meanings. These maps not only help visualize how meanings vary across languages, but can also be used to determine how languages vary across time. Thus, Georgakopoulos and Polis (2021) created diacronic semantic maps both for Ancient Egyptian and Ancient Greek. They argue that these maps properly reflect the expected semantic changes that happened during the chosen period of time.

5. NLP for Coptic

Even though Coptic can be considered a later stage of Ancient Egyptian, it has important differences with respect to Classical and Late Egyptian (Bard, 2005). This leads to a different set of problems when using NLP techniques with the language. One of these differences is that Coptic is no longer written in hieroglyphs, as it uses a modified version of the Greek alphabet instead. This leads to transliteration no longer being an issue, as there is a one-to-one correspondence between symbols and phonemes.

Another factor is that the morphology of the language went through several major changes. One example of this is the difference in the usage of affixes along with a huge influx of loanwords from Greek, which did not always adapt to the Coptic morphology (Kramer, 2006; Zeldes and Schroeder, 2016). An example on how this affects the design of NLP tools is with segmentation, especially when attempting to detect the language origin of a word.

A lot of documents from early Christianity were written in Coptic and the Coptic Orthodox Church still uses the language during mass. This means that there are more well-preserved texts in Coptic than in Ancient Egyptian. Thus, the contents of these texts tend to attract more attention from a wider variety of scholars such as those in Christian theology and related fields.

5.1. Morphological Analysis

Smith and Hulden (2016) did morphological analysis on Sahidic Coptic, one of the dialects of Coptic. They consider that a good model could be a transducer as it is mainly a prefixing language save for a few notable exceptions. Their testing set was composed of over a hundred words and had a recall slightly lower than 0.95. They think that their work could be useful for teaching the Coptic grammar and note that it could help study the larger Coptic texts. However, they make no mention on whether their model would need major modifications to consider other dialects, only stating that increasing the coverage of their analyser would need more lexicographical work.

Meanwhile, Ashton (2012) use a combination of a context-free grammar and transducer to model a smaller-scale morphological phenomenon, namely, second position clitics in Sahidic Coptic. They base the rules for their grammar in the linguistic literature. They do not provide any implementation or experimental results, as they note that an actual implementation of their system would be complicated from a technical point of view.

5.2. Named Entity Recognition

Yousef et al. (2023) combined out-of-the-box named entity recognition (NER) systems with transformer-based architectures for text alignment. Their system worked reasonably well for Ancient Greek and Latin versions of the Bible. However, they note that this approach did not work when dealing with Coptic versions of the same texts.

On the other hand, Khakhmovich et al. (2020) propose to use cross-lingual transliteration with transformer-based models as a way to tackle out-of-vocabulary terms, using Coptic as an example among other languages.

[21] http://github.com/estibalizifranjo/hieroglyphs

5.3. The Coptic Scriptorium and Universal Dependencies

As was mentioned in Section 3, the Coptic Scriptorium (Schroeder and Zeldes, 2016) is a corpus that had at its release a little less than 60 thousand words available. Several tools have been developed to be used along with it, which we will talk about in the rest of this section.

Zeldes and Abrams (2018) considered that the creation of a treebank compatible with the Universal Dependency (UD)[22] (de Marneffe et al., 2021) annotation scheme would be an important addition to the study of Coptic in general. They decided to work with the Coptic Scritptorium corpus due to it being freely available and also that the automatic segmentation achieves a very high precision score, which means that it can be considered a gold standard. They mainly decided to follow two main principles: when possible their notation should be compatible with the previous literature in the field and they would try to keep the notation in line with the practices in Hebrew and Arabic, which come from the same language family. When testing their treebank against expert human annotators, they got an agreement of over 95%. The agreement dropped to slightly over 85% when compared to undergraduate students. This was the first treebank built for the Egyptian language subfamily.

Another tool for the Coptic Scriptorium came in the form of a pipeline for NLP analysis. Zeldes and Schroeder (2016) created an online tool that automates several tasks, namely segmentation, normalization, tagging and lemmatization, detection of language of origin, and parsing.

For the segmentation task they selected around 180 rules and created a model that determined the priority order of the rules through 10-fold cross-validation. The accuracy of this model was slightly higher than 0.9. In the normalization stage, they had to consider the use of diacritics, spelling variations, and abbreviations. For this task, they used a combination of a predetermined list of common variations and a learnt list of the use of diacritics and capitalization. This model had an accuracy of 0.98. For part-of-speech tagging and lemmatization, they used an algorithm called TreeTagger (Schmid, 1999) and achieved accuracies of 0.95 and 0.97, respectively. As for determining whether the language of the text was Coptic, they had an accuracy of over 0.93. Finally, the parsing section has a preliminary version of the model of the paper from Zeldes and Abrams (2018) mentioned previously in this Section, which achieves an accuracy of 0.87.

Each of the components on the paper by Zeldes and Schroeder (2016) can be used either on their own or as part of a pipeline and can be accessed both at the author's website[23] or as part of the Coptic Scriptorium project[24].

As part of UD, the Coptic Scritorium has also been used for other projects. One of these was the the second shared task of SIGMORPHON 2019 (McCarthy et al., 2019), which was on morphological analysis given a word's context. The winning team (Straka et al., 2019) used an ensemble of nine LSTM (Hochreiter and Schmidhuber, 1998) models using BERT (Devlin et al., 2019). They also joined subcorpora from different languages. Their model achieved the highest performance on the Coptic subcorpus, with a lemma accuracy of 0.97 and a morpheme accuracy of 0.96.

Other projects in which the UD version of the Coptic Scriptorium has been used are multilingual dependency parsing (Dehouck and Denis, 2019; Choudhary, 2021; Choudhary and O'riordan, 2023), morphological tagging (Chakrabarty et al., 2019), studying the order of cosisters[25] (Dyer, 2018), studying information-theoretic locality properties of trees (Futrell, 2019), developing a multilingual categorical grammar (Tran and Miyao, 2022), as well as studying whether quantitative laws of language hold (Berdicevskis, 2021). We don't go into technical details of these approaches as Coptic is not a central part of any of these papers.

Finally, it has also been used as part of a study on the quality of the different treebanks of UD (Kulmizev and Nivre, 2023). While the Coptic treebank scores well in most of the metrics investigated in that paper, the authors note that it is one of the bottom three treebanks in terms of variability as defined by Swayamdipta et al. (2020).

6. Summary & Conclusion

The use of NLP methods on Ancient Egyptian is useful as it can help us gain insights both from a linguistic and from a historical standpoint. However, the advances in this field of research have been sparse through time. Polis et al. (2013) and Nederhof and Rahman (2015) consider that this has been in good part due to the lack of annotated text. They also note that most attempts are trying to generalize over large periods of time even when taking into account divisions such as Middle and Old Egyptian.

Another notable thing is that most papers have focused on Coptic. This is understandable as its inclusion in the UD project means that it has access to a wide array of tools that are being developed

[22] https://universaldependencies.org/

[23] https://corpling.uis.georgetown.edu/coptic-nlp/

[24] https://copticscriptorium.org/

[25] Defined in that paper as "sister constituents of the same syntactic form on the same side of their head".

with this project in mind. However, this tends to shift attention from the other stages of Ancient Egyptian, with Demotic being the most affected.

In their 2017 talk, Polis and Razanajao (2017) note that more interaction between projects could be useful, not only in the field of computational linguistics, but in Egyptology as a whole. This is especially important as most projects use either the same datasets or the same objects, but end up having their own systems and annotation schemes that are not compatible with each other. An example they give is that of a statue with inscriptions. The artifact itself has value for some researchers, while the kind of object or its inscriptions might be of interest to others. They also note that, while some researchers might be interested in the location and the layout of the text, some others might be just interested in the text itself or even in just the content. They mention that there is a current collaborative project called THOT (Dils et al., 2018) that aims to be a bridge for these areas of study. While the project does not have any sort of connection to the actual databases, their website has a roadmap to show how it will grow in the future.

This area of research appears to be approached by a very limited amount of researchers. However, some of these research groups appear to be growing, such as the one dedicated to the Ramses corpus, the evolution of which can be seen in Polis et al. (2013), Polis et al. (2015), and Polis and Razanajao (2017). We hope that this work will bring about a larger interest and allow for fruitful collaborations between the fields of NLP and Egyptology.

As a final note, an interesting thing would be to compare and contrast the NLP advances that have been done in other ancient languages, such as Sumerian, Ancient Greek, Sanskrit, etc. This could show how the advances in these different languages have affected or influenced each other. Even though some of the papers that we have mentioned so far did show this, most did not. A development in this direction comes from an NLP package called The Classical Language Toolkit (Johnson et al., 2021). It has tools for several ancient languages and even provides access to corpora for several of them, including the Coptic Scriptorium corpora mentioned in Section 3. This package could help encourage more research on these languages, which will help in turn gain important insights into our past.

7. Bibliographical References

Neil Ashton. 2012. Second position clitics and monadic second-order transduction. In *Proceedings of the Workshop on Applications of Tree Automata Techniques in Natural Language Processing*, pages 31–41, Avignon, France. Association for Computational Linguistics.

Kathryn A. Bard. 2005. Egyptian language and writing. In *Encyclopedia of the Archaeology of Ancient Egypt*, pages 325–328. Routledge. Google-Books-ID: MH7sAgAAQBAJ.

François Barthélemy and Serge Rosmorduc. 2011. Intersection of multitape transducers vs. cascade of binary transducers: The example of egyptian hieroglyphs transliteration. In *Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing*, pages 74–82. Association for Computational Linguistics.

Aleksandrs Berdicevskis. 2021. Successes and failures of menzerath's law at the syntactic level. In *Proceedings of the Second Workshop on Quantitative Syntax (Quasy, SyntaxFest 2021)*, pages 17–32, Sofia, Bulgaria. Association for Computational Linguistics.

Branislav Bédi, Belinda Chiera, Cathy Chua, Brynjarr Eyjólfsson, Manny Rayner, Catherine Orian Weiss, and Rina Zviel-Girshin. 2022. Using LARA to create annotated manuscripts and inscriptions for museums: an initial feasibility study. In Birna Arnbjörnsdóttir, Branislav Bédi, Linda Bradley, Kolbrún Friðriksdóttir, Hólmfríður Garðarsdóttir, Sylvie Thouësny, and Matthew James Whelpton, editors, *Intelligent CALL, granular systems and learner data: short papers from EUROCALL 2022*, 1 edition, pages 18–23. Research-publishing.net.

Abhisek Chakrabarty, Akshay Chaturvedi, and Utpal Garain. 2019. Neumorph: Neural morphological tagging for low-resource languages—an experimental study for indic languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(1).

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Chinmay Choudhary. 2021. Improving the performance of UDify with linguistic typology knowledge. In *Proceedings of the Third Workshop on Computational Typology and Multilingual NLP*, pages 38–60, Online. Association for Computational Linguistics.

Chinmay Choudhary and Colm O'riordan. 2023. Multilingual end-to-end dependency parsing with linguistic typology knowledge. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 12–21, Dubrovnik, Croatia. Association for Computational Linguistics.

Mathieu Dehouck and Pascal Denis. 2019. Phylogenic multi-lingual dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 192–203, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William Dyer. 2018. Integration complexity and the order of cosisters. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 55–65, Brussels, Belgium. Association for Computational Linguistics.

Richard Futrell. 2019. Information-theoretic locality properties of natural language. In *Proceedings of the First Workshop on Quantitative Syntax (Quasy, SyntaxFest 2019)*, pages 2–15, Paris, France. Association for Computational Linguistics.

Thanasis Georgakopoulos and Stéphane Polis. 2018. The semantic map model: State of the art and future avenues for linguistic research. *Language and Linguistics Compass*, 12(2):e12270. E12270 LNCO-0727.R1.

Thanasis Georgakopoulos and Stéphane Polis. 2021. Lexical diachronic semantic maps: Mapping the evolution of time-related lexemes. *Journal of Historical Linguistics*, 11(3):367–420.

Stéphanie Gohy, Benjamin Martin, and Polis Stéphane. 2013. Automated text categorization in a dead language. the detection of genres in late egyptian. In *Texts, Languages & Information Technology in Egyptology. Selected papers from the meeting of the Computer Working Group of the International Association of Egyptologists (Informatique & Égyptologie), Liège, 6-8 July 2010*, Aegyptiaca Leodiensia. Presses Universitaires de Liège. Backup Publisher: F.R.S.-FNRS - Fonds de la Recherche Scientifique.

Sepp Hochreiter and Jürgen Schmidhuber. 1998. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Janice Kamrin. 2004. *Ancient Egyptian Hieroglyphs: A Practical Guide - A Step-by-Step Approach to Learning Ancient Egyptian Hieroglyphs*. Harry N. Abrams. Google-Books-ID: JsWZQgAACAAJ.

Aleksandr Khakhmovich, Svetlana Pavlova, Kira Kirillova, Nikolay Arefyev, and Ekaterina Savilova. 2020. Cross-lingual named entity list search via transliteration. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4247–4255, Marseille, France. European Language Resources Association.

Ruth Kramer. 2006. Root and pattern morphology in coptic: Evidence for the root. In *Proceedings of the 36th Annual Meeting of the North East Linguistic Society*, volume 2. University of Massachussets Amherst.

Artur Kulmizev and Joakim Nivre. 2023. Investigating UD treebanks via dataset difficulty measures. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1076–1089, Dubrovnik, Croatia. Association for Computational Linguistics.

Jackson L. Lee, Lucas F.E. Ashby, M. Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D. McCarthy, and Kyle Gorman. 2020. Massively multilingual pronunciation modeling with WikiPron. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4223–4228, Marseille, France. European Language Resources Association.

C.K. Maisels. 1998. *Near East: Archaeology in the 'Cradle of Civilization'*. The experience of archaeology. Routledge.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

Georg Möller. 1919. Die buchschrift der alten Ägypter. In *Zeitschrift des Deutschen Vereins für Buchwesen und Schrifttum*, volume 2, pages 73–79. Verlag des Deutschen Vereins für Buchwesen und Schrifttum.

Mark Jan Nederhof. 2009. Automatic alignment of hieroglyphs and transliteration. In *Information Technology and Egyptology in 2008: Proceedings of the meeting of the Computer Working Group of the International Association of Egyptologists*, 2. Gorgias Press. Accepted: 2011-01-07T14:05:02Z ISSN: 1943-9369.

Mark-Jan Nederhof and Fahrurrozi Rahman. 2015. A probabilistic model of ancient egyptian writing. In *Proceedings of the 12th International Conference on Finite-State Methods and Natural Language Processing 2015 (FSMNLP 2015 Düsseldorf)*. Association for Computational Linguistics.

Mark-Jan Nederhof and Fahrurrozi Rahman. 2017. A probabilistic model of ancient egyptian writing. *Journal of Language Modelling*, 5(1):131–163.

Serge Rosmorduc. 2009. Automated transliteration of egyptian hieroglyphs. In Nigel Strudwick, editor, *Information Technology and Egyptology in 2008*, pages 167–182. Gorgias Press.

Serge Rosmorduc. 2015. Computational linguistics in egyptology. *UCLA Encyclopedia of Egyptology*, 1(1).

Daniel Smith and Mans Hulden. 2016. Morphological analysis of sahidic coptic for automatic glossing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2584–2588. European Language Resources Association (ELRA).

Milan Straka, Jana Straková, and Jan Hajic. 2019. UDPipe at SIGMORPHON 2019: Contextualized embeddings, regularization with morphological categories, corpora merging. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 95–103, Florence, Italy. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9275–9293, Online. Association for Computational Linguistics.

Tu-Anh Tran and Yusuke Miyao. 2022. Development of a multilingual CCG treebank via Universal Dependencies conversion. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5220–5233, Marseille, France. European Language Resources Association.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Philipp Wiesenbach and Stefan Riezler. 2019. Multi-task modeling of phonographic languages: Translating middle Egyptian hieroglyphs. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.

Tariq Yousef, Chiara Palladino, Gerhard Heyer, and Stefan Jänicke. 2023. Named entity annotation projection applied to classical languages. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 175–182, Dubrovnik, Croatia. Association for Computational Linguistics.

8. Language Resource References

Branislav Bédi, Hakeem Beedar, Belinda Chiera, Nedelina Ivanova, Christèle Maizonniaux, Neasa Ní Chiaráin, Manny Rayner, John Sloan, and Ghil'ad Zuckermann. 2022. Using LARA to create image-based and phonetically annotated multimodal texts for endangered languages. In *Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 68–77, Dublin, Ireland. Association for Computational Linguistics.

Walter E. Crum. 1939. *A Coptic Dictionary*. Oxford University Press.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Peter Dils, Silke Grallert, Ingelore Hafemann, Stéphane Polis, Lutz Popko, Vincent Razanajao, Simon Schweitzer, and Daniel Werning. 2018. Thot - thesauri and ontology for ancient egyptian resources.

Frank Feder, Maxim Kupreyev, Emma Manning, Caroline T. Schroeder, and Amir Zeldes. 2018. A linked Coptic dictionary online. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 12–21, Santa Fe, New Mexico. Association for Computational Linguistics.

Andrew Glass, Jorke Grotenhuis, Mark-Jan Nederhof, Stephane Polis, Serge Rosmorduc, and Daniel A Werning. 2021. Additional control characters for ancient egyptian hieroglyphic texts. Accessed: 2024-02-15.

Estíbaliz Iglesias-Franjo and Jesús Vilares. 2020. Searching four-millenia-old digitized documents: A text retrieval system for egyptologists. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 22–31. Association for Computational Linguistics.

Janet H Johnson, editor. 2001. *The Demotic Dictionary of the Oriental Institute of the University of Chicago*. The Oriental Institute, University of Chicago.

Kyle P. Johnson, Patrick J. Burns, John Stewart, Todd Cook, Clément Besnier, and William J. B. Mattingly. 2021. The Classical Language Toolkit: An NLP framework for pre-modern languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 20–29, Online. Association for Computational Linguistics.

Mark-Jan Nederhof, Stéphane Polis, Serge Rosmorduc, and Simon Schweitzer. 2019. Unicode control characters for ancient egyptian. In *12th International Congress of Egyptologists*. IFAO, Cairo, Egypt.

Sebastian Nordhoff and Thomas Krämer. 2022. IMTVault: Extracting and enriching low-resource language interlinear glossed text from grammatical descriptions and typological survey articles. In *Proceedings of the 8th Workshop on Linked Data in Linguistics within the 13th Language Resources and Evaluation Conference*, pages 17–25, Marseille, France. European Language Resources Association.

Stéphane Polis, Luc Desert, Peter Dils, Jorke Grotenhuis, Vincent Razanajao, Tonio Sebastian Richter, Serge Rosmorduc, Simon D. Schweitzer, Daniel A. Werning, and Jean Winand. 2021. The thot sign list (tsl). an open digital repertoire of hieroglyphic signs. *Égypte nilotique et méditerranéenne*, 14.

Stéphane Polis, Anne-Claude Honnay, and Jean Winand. 2013. Building an annotated corpus of late egyptian. the ramses project: Review and perspectives. In *Texts, languages & information technology in egyptology: selected papers from the meeting of the Computer Working Group of the International Association of Egyptologists*, Aegyptiaca Leodiensia, pages 25–44. Presses Universitaires de Liège. OCLC: 843421912.

Stéphane Polis and Vincent Razanajao. 2017. Ancient egyptian philology: The digital turn. current projects and future perspectives for the study of ancient egyptian texts. In *Global Philology Open Conference*. Mondes anciens.

Stéphane Polis, Serge Rosmorduc, and Jean Winand. 2015. Ramses goes online. an annotated corpus of late egyptian texts in interaction with the egyptological community. International Congress of Egyptologists XI.

Helmut Schmid. 1999. Improvements in part-of-speech tagging with an application to german. In Susan Armstrong, Kenneth Church, Pierre Isabelle, Sandra Manzi, Evelyne Tzoukermann, and David Yarowsky, editors, *Natural Language Processing Using Very Large Corpora*, Text, Speech and Language Technology, pages 13–25. Springer Netherlands.

Caroline T. Schroeder and Amir Zeldes. 2016. Coptic SCRIPTORIUM: A corpus, tools, and methods for corpus linguistics and computational historical research in ancient egypt. In *White Paper*. University of the Pacific.

Stephan J. Seidlmayer. 2011. Handbuch zur benutzung des thesaurus linguae aegyptiae (TLA). Berlin-Brandenburg Academy of Sciences and Humanities.

Laura Slaughter, Luis Morgado Da Costa, So Miyagawa, Marco Büchler, Amir Zeldes, and Heike Behlmer. 2019. The making of Coptic Wordnet. In *Proceedings of the 10th Global Wordnet Conference*, pages 166–175, Wroclaw, Poland. Global Wordnet Association.

Jörg Tiedemann. 2016. OPUS – parallel corpora for everyone. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation: Projects/Products*, Riga, Latvia. Baltic Journal of Modern Computing.

Amir Zeldes and Mitchell Abrams. 2018. The coptic universal dependency treebank. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 192–201. Association for Computational Linguistics.

Amir Zeldes and Caroline T. Schroeder. 2016. An NLP pipeline for coptic. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 146–155. Association for Computational Linguistics.

A. Writing Systems

In this appendix we illustrate what the writing systems of the different variations of Ancient Egyptian looked like through a few examples.

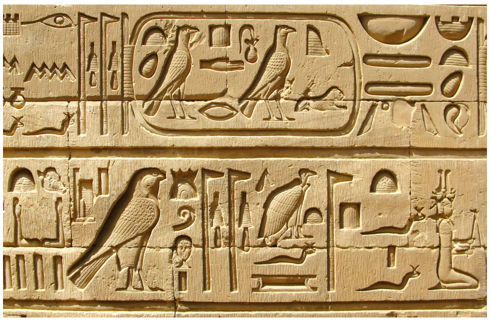


Figure 2: An example of hieroglyphs from the Temple of Kom Ombo in Egypt. Picture taken from Encyclopaedia Britannica. This temple was built during the Ptolemaic Dynasty from 180 to 47 BC.
Copyright: Icon72/Dreamstime.com.
https://www.britannica.com/topic/hieroglyph#/media/1/265009/118144 (Accessed March 30, 2024)

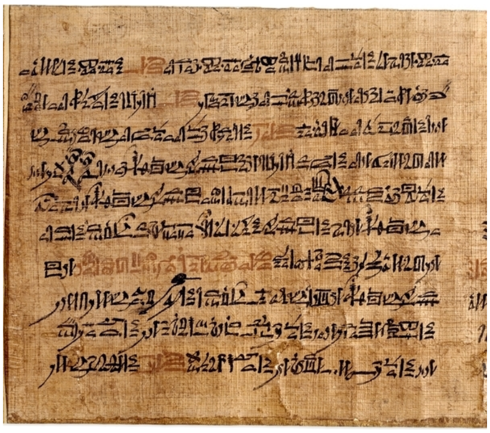


Figure 3: A sheet in hieratic from the Papyrus D'Orbine. It contains part of the Tale of Two Brothers. This document was written during the 19th Dynasty, circa 1185 BC.
Copyright: Image in the public domain.
https://commons.wikimedia.org/wiki/File:Tale_of_two_brothers.jpg (Accessed March 30, 2024)

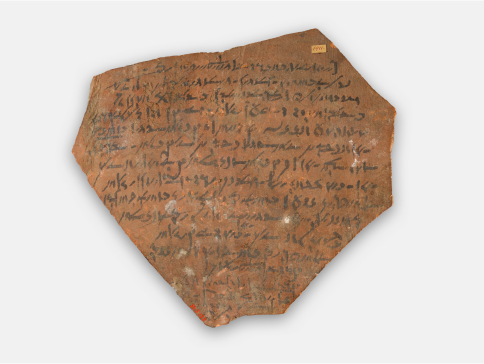


Figure 4: A text written in demotic script, from the Ptolemaic period (127 BC). It is an oath to the god Hathor denying the author's involvement in a cloths-theft.
Copyright: Rogers Fund, 1921. Image available under the Creative Commons CC0 1.0 Universal Public Domain Dedication.
https://commons.wikimedia.org/wiki/File:Demotic_Temple_Oath_MET_LC-21_2_122_EGDP023779.jpg (Accessed March 30, 2024)

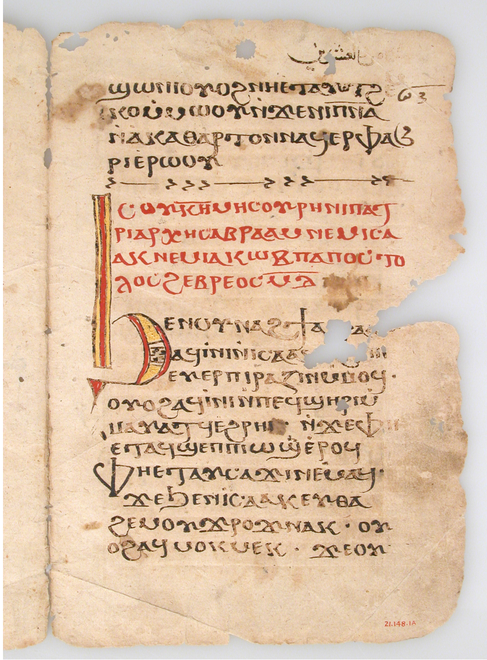


Figure 5: A page from a manuscript in Coptic. It is from sometime between the 6th and 14th centuries.
Copyright: Rogers Fund, 1921. Image available under the Creative Commons CC0 1.0 Universal Public Domain Dedication.
https://commons.wikimedia.org/wiki/File:Leaves_from_a_Coptic_Manuscript_MET_sf21-148-1as3.jpg (Accessed March 30, 2024)

Towards a Readability Formula for Latin

Thomas Laurs
Georg-August Universität Göttingen
th.laurs@gmail.com

Abstract

This research focuses on the development of a readability formula for Latin texts, a much-needed tool to assess the difficulty of Latin texts in educational settings. This study takes a comprehensive approach, exploring more than 100 linguistic variables, including lexical, morphological, syntactical, and discourse-related factors, to capture the multifaceted nature of text difficulty. The study incorporates a corpus of Latin texts that were assessed for difficulty, and their evaluations were used to establish the basis for the model. The research utilizes natural language processing tools to derive linguistic predictors, resulting in a multiple linear regression model that explains about 70% of the variance in text difficulty. While the model's precision can be enhanced by adding further variables and a larger corpus, it already provides valuable insights into the readability of Latin texts and offers the opportunity to examine how different text genres and contents influence text accessibility. Additionally, the formula's focus on objective text difficulty paves the way for future research on personal predictors, particularly in educational contexts.

Keywords: Readability, Latin, Readability Formula, Linguistic Predictors

1. Introduction

1.1 Readability and Text Comprehension

A method for assessing the difficulty of Latin texts remains a desideratum even though having an objective and precise understanding of the complexity of Latin texts offers numerous advantages in both school and university settings. This knowledge is beneficial for selecting appropriate texts, not only for assessments but also for classroom instruction. It enables textbook authors to craft texts with a steadily increasing level of difficulty, and after the work with the textbook, instructors can use a readability formula to choose suitable texts from authentic Latin authors. The knowledge of text difficulty is especially crucial when it comes to selecting examination texts. This is particularly significant in times of standardized testing, where objective text selection stands as a critical criterion.

Text difficulty, often called readability, is a measure of how smoothly processes of text comprehension can unfold. These processes are determined by both textual features and reader attributes (Friedrich, 2017). Textual features can be divided into two distinct categories. On one hand, texts exhibit a surface structure, encompassing all easily quantifiable linguistic features. On the other hand, texts possess a deep structure, comprising content-related and stylistic features of the text, the translation of which into a numerical value is relatively complex (Groeben, 1982). However, it is essential to note that the boundaries between surface and deep structure are not strictly delineated because some elements of the deep structure can also be calculated objectively. While textual features remain constant within the same text, reader attributes vary, explaining why different readers perceive the same text as more or less difficult. This variation is due to differences in the most important reader attributes, such as intelligence, interest, and prior knowledge (Rost, 2018).

To accurately measure text difficulty, understanding the processes involved in text comprehension is crucial. In general, it can be said that the reader decodes the linguistic information of the text's surface, which includes morphology and syntax, and thus creates a list of propositions at the level of the so-called text base. Subsequently, these propositions are enriched through automatically occurring inferences, resulting in an initial, yet not fully coherent network of propositions. Finally, through actively drawn inferences, reorganization, and reinstatement, a self-contained propositional network is established (the so-called construction-integration model of Kintsch, 1988). Even though the processes of text comprehension for Latin, that might differ from modern languages since being a dead language, have not been extensively researched, this model can be posited for Latin as well due to its generality.

1.2 Phases of Readability Research

In order to develop a metric for predicting the difficulty of texts, readability research has, for about a century, developed various methods, all of which can fundamentally be traced back to the same scheme: (α) Initially, a corpus of texts, whose difficulty has been assessed using a criterion (e.g., a reading test, Cloze test, expert judgment, Common European Framework of Reference for Languages (CEFR)), is gathered. (β) From these texts, linguistic variables are collected. (γ) Finally, the relationships between the predictors and the criterion are statistically modeled (François and Fairon, 2012).

At the beginning of readability research, researchers initially focused on a few linguistic variables, primarily word length as a proxy of vocabulary frequency and sentence length as a proxy for syntactic complexity. Of particular significance in this context are the formulas of Flesch (1948) and Dale and Chall (1948). Both selected a corpus of almost 400 texts. As a criterion, the difficulty was determined through a reading test. Both formulas were established through linear regression and incorporate the two mentioned linguistic variables.

Because these two variables could seem to be too superficial to determine something as complex as the readability of a text, strong criticism of existing formulas has been voiced since 1979 (inter alia Kintsch and Vipond, 1979; Selzer, 1981; Groeben, 1982). Researchers at that time have employed predictors, that were intended to better represent the processes of text comprehension, such as the number of propositions, inferences, or reinstatements, and other deep structural linguistic variables. However, determining these predictors not only requires a considerable effort but is often non-objective. Furthermore, the novel variables and formulas cannot predict text difficulty better than traditional approaches (Kintsch and Miller, 1984).

In recent years, researchers have increasingly turned towards methods of computational linguistics. This allows them to significantly expand the corpus of texts. The difficulty of the texts is usually not assessed by subjects, but often the CEFR is used as a criterion. Machine learning can also be used to rapidly create complex models with numerous linguistic variables (Benjamin, 2012; Vajjala, 2022).

However, there is currently no state-of-the-art readability model for Latin. While some readability formulas exist (e.g., Bayer, 2003 or Gruber-Miller and Mulligan, 2022), their formulas are either based more on theoretical considerations than empiricism or comprise only one linguistic category. In Bayer's formula, a corpus of Latin texts whose difficulty was assessed by a criterion is missing. And Gruber-Miller and Mulligan focused their study only on lexical variables. The goal of this work is to propose a first readability model that follows the established methods of readability research: The difficulty of 67 Latin texts was estimated by students; nearly 200 linguistic variables were calculated using NLP-tools; via stepwise multiple linear regression, a readability model was created to provide a more holistic understanding of Latin text complexity.

2. Empirical Study

2.1 Corpus

There is currently no corpus of Latin texts whose difficulty has been estimated by using an adequate criterion. Since cloze tests and reading tests are not feasible for Latin, we created a questionnaire with a Likert scale, that consisted of 50 items. Bachelor and master students had to read and translate Latin texts and then assessed their difficulty using this questionnaire. They had learned Latin as a historic language in a traditional way. The items of the questionnaire were developed with reference to the theory of the processes of text comprehension presented above and were subsequently analyzed statistically. In total, the 13 best items were retained, which exhibit high discriminatory power and are overall unidimensional, i.e., they all load onto the same factor in the Principal Component Analysis (PCA). All the items are listed in table 1.

In addition to the items based on text comprehension, six additional questions were included to assess the personal knowledge and interests of the participants. After all, personal predictors also influence individual perceptions of difficulty. To eliminate this confounding factor, the same Likert scale was used to gather information about how well the students are versed in vocabulary, grammar, ancient culture and mythology, how well their knowledge is about the given Latin author or literary genre, as well as their level of interest in Latin literature and the duration of their engagement with Latin texts. All six factors exhibited slight correlations with the participants' difficulty assessments, with the strongest correlations observed for knowledge of author and genre (r = 0.35) and grammar (r = 0.27). As a result, these confounding factors were removed, and, after transforming the modified values onto a 1 to 10 scale, the adjusted difficulty of the texts was obtained. To sum it up, the Latin text of the corpus got their respective difficulty score through the individual difficulty estimations of the students guided through the questionnaire.

#	Question
1	The meanings of most words became clear to me quickly.
2	The sentences had a straightforward syntactic structure.
3	I found it challenging to anticipate how the sentence would continue syntactically.
4	The text contradicted some of the expectations I had formed while reading.
5	I had to frequently backtrack in the text to understand what was being conveyed.
6	Throughout the reading, I had all the necessary information in mind to comprehend the text.
7	At various points, I wished for greater precision in what was meant.
8	Providing a summary of the text would be easy for me.
9	I found it difficult to differentiate between what was important and unimportant in the text.
10	The text was written vividly.
11	I struggled to form a mental image of the content while reading.
12	I found the text to be comprehensible.
13	All in all, the text was easy to understand.

Table 1: Items of the questionnaire

Table 2 includes a selection of five text passages along with their difficulty scores. All in all, 67 Latin texts were assessed by students, 40 prose texts and 27 from poetry, comprising a range of diverse classical authors. The texts had a length of ca. 180 words.

Text passage	Difficulty Score
Pliny 7.19	1.12
Ov. *Met.* 1.283–296	2.54
Verg. *Aen.* 3.147–178	3.29
Livy 44.22.1–8	4.78
Lucan 9.1–33	6.46

Table 2: Difficulty scores of selected texts

2.2 Predictors

Nearly 200 linguistic variables from the areas of Lexicon, Morphology, Discourse, and Syntax were examined. It is not possible to describe all the variables at this point. Therefore, the domains of the linguistic variables will be outlined briefly, and selected linguistic variables will be described.

2.2.1 Lexicon and Semantics

For Latin, the area of Lexicon and Semantics is particularly crucial. Unlike native speakers, Latin learners must actively acquire vocabulary. If they lack knowledge of the words or cannot retrieve them quickly enough while reading, text comprehension is severely impeded.

The investigation of Lexicon and Semantics is divided into four major categories: (1) word length, (2) word frequency, (3) lexical density, and (4) polysemy.

2.2.1.1 Word Length

Word length is one of the most used variables in readability research. On the one hand, it is easy to calculate, and on the other hand, it serves as a proxy for word frequency (Berendes et al., 2018), because shorter words are more frequent and thus can be understood better by readers (Zipf, 1935). Besides average word length itself, measures like the percentage of monosyllabic words – that can be prepositions, pronouns, verb forms etc. – are added.

2.2.1.2 Word Frequency

Since word length is merely a proxy for word frequency, it is advisable to directly calculate word frequency. Word frequency can be indirectly calculated by examining the percentage of words that do not appear in a list of the most common Latin words (e.g., DCC Latin Core Vocabulary). Alternatively, direct calculations are also possible by determining the number of both lemmas and distinct word forms (i.e. types). In this context, so-called stop words can be excluded, i.e., words that do not significantly contribute to the content of a text, such as conjunctions, etc. (Vogel and Washburne, 1928; McNamara et al., 2014). To ascertain the number of lemmas and the most common Latin words, a corpus comprising texts from Plautus to Augustine was amassed, totaling more than 2 million words. Subsequently, the respective variables of word frequency were computed based on this corpus.

2.2.1.3 Lexical Density

The standard measure for Lexical Density is the Type-Token Ratio (TTR) along with its various calculation methods that aim to minimize the influence of text length (Berendes et al., 2018). Additionally, other measures include the ratio of content words to function words or the curve length R, which is obtained from a rank-frequency distribution by taking the Euclidean distances between adjacent points (Mikros and Voskaki, 2021, following Kubát et al., 2014). This area also encompasses the analysis of Parts of Speech (POS), i.e., examining the ratio of nouns to verbs in a text (Xia et al., 2016).

2.2.1.4 Polysemy

Furthermore, a consideration of polysemy is of paramount importance, especially for Latin, as Latin words are often polysemous and can pose greater difficulties for learners because they may not immediately grasp the meaning, that is correct in each context (McNamara et al., 2014). Polysemy can be determined using the Latin WordNet (LWN). As LWN is not complete, words not covered by the resource were omitted from calculation. Additionally, the number of polysemies can also be determined using the OLD (Oxford Latin Dictionary). The number of meanings given by the OLD of the most important content were stored in a database. From that, the score of polysemy was calculated.

2.2.2 Morphology

As a highly inflected language, Latin, in contrast to English, offers a wider range of difficulties in morphology. Therefore, the occurrence of specific verb forms – ordered by person and number, tense, mood, and voice – as well as the cases of nouns were examined.

2.2.3 Syntax

In the realm of syntax, calculations were carried out in the domains of (1) sentence length, (2) sentence structure, (3) sentence composition, (4) discontinuous noun phrases, and (5) syntactic phenomena.

Sentence length is the traditional measure most frequently used in readability literature (Gray and Leary, 1935; Hancke et al. 2012). In addition to sentence length, the clause length is also significant.

Syntax in Latin places a greater emphasis on word order than in English. This is because the word order in Latin is relatively free. For example, the number of words before the predicate of the main clause or the number of instances where the object precedes the subject of the clause were examined.

Latin prose in particular tends to compose texts in nested complex sentences. One measure to capture this is dependency length, which is also used as a measure of syntactic complexity by Futrell et al. (2015) or Berendes et al. (2018).

Discontinuous noun phrases, also called *hyperbata*, are typical for Latin, especially for Latin poetry, and quite frequent (Haug, 2017). Because of their complexity, they cannot be determined precisely enough by NLP tools, that's why they were calculated manually. The other variables in the syntactic domain were calculated via latinCy, v. infr. Additionally, typical syntactic phenomena such as *Accusativus cum Infinitivo* (AcI) or Gerundive were also manually calculated.

2.2.4 Discourse Variables

In addition to these surface-level text variables, linguistic variables of the deep structure known as discourse-related variables can be considered. The primary goal is to measure the coherence of a text, that means that the text is referring to its own content and connecting the content logically through connectors, pronouns, or co-references. We can calculate that by instances of identical words or lemmas in consecutive sentences (Todirascu et al., 2013; McNamara et al., 2014). Apart from co-reference, latent semantic analysis (LSA) provides another measure of sentence overlap. Essentially, it involves converting the sentences of a text into vectors and determining their similarity using the cosine measure (François and Fairon, 2012).

2.3 Results

The individual predictors were determined using Natural Language Processing (NLP) techniques. Pre-built tools were employed for this purpose, including the Classical Language Toolkit (CLTK), Stanza, and spaCy (latinCy). However, especially in the realm of syntax, these programs are not yet precise enough (Burns, 2023). Therefore, caution is advised when interpreting the results of the syntactic variables. In addition, some important Latin predictors have been determined manually, including the number of hyperbata (discontinuous noun phrases) and the number of specific syntactic phenomena such as AcI, Ablative Absolute, Gerundives, and so on. The following table 3 contains 20 selected linguistic variables with their correlation coefficients: variables 1–9 come belong to lexicon and semantics, 10–12 to morphology, 13–17 to syntax, and 18–20 to discourse.

#	Description	r
1	Word lengths in letters	.07
2	Percentage of one syllable words	-.33
3	Inverse lemma frequency	.37
4	Frequency of word forms, without stop words, sorted by rank	.23
5	Percentage of words outside a list of the most frequent 750 Latin words	.55
6	Type token ratio, without stop words	.22
7	Ratio of content words to function words	.42
8	Ratio of nouns to all words	.41
9	Average number of polysemes, without stop words, according to the Latin WordNet	-.05
10	Instances of verbs in 3rd singular	.27
11	Instances of verbs in 2nd plural	.25
12	Instances of verbs in pluperfect	-.22
13	Sentence lengths in words	.11
14	Sentence depth, divided by number of t-units	-.05
15	Ratio of finite subclauses to all subclauses	-.30
16	Number of interlaced hyperbata	.54
17	Combination of the easiest syntactic phenomena	-.42
18	Number of connectors	-.25
19	Ratio of pronouns to all words	-.31
20	LSA	-.21

Table 3: Selected linguistic variables with correlation coefficients (r)

The impact on text difficulty is generally greater for lexical variables than for syntax. Word frequency and lexical density, in particular, exhibit a high correlation. Furthermore, these variables tend to yield higher scores in poetic texts. Consequently, it is unsurprising that poetic texts generally receive higher difficulty scores. Contributing to this higher difficulty are also the number of discontinuous noun phrases, which are more prevalent in poetic texts. It is noteworthy that the two standard variables of classical readability studies, word and sentence length, do not exhibit significant correlations with text difficulty in Latin. When examining correlations separately for prose and poetic texts, it becomes apparent that lexical variables exert a greater influence on text difficulty in poetic texts, whereas syntactic variables are more important for computing the difficulty of prose texts.

To model the relationship between linguistic variables and the difficulty of individual texts, a multiple linear regression analysis was conducted as a statistical model. The selection of appropriate variables is not trivial. A stepwise regression analysis was performed: initially, a regression was created with only one parameter, the highest correlated variable (#5). Subsequently, from the remaining variables, the one that resulted in the lowest root-mean-square deviation in a 10-fold cross-validation was added to the model, while all p-values should not fall below the level of significance. This process continued until no significant p-values were obtained. Since the text difficulty here is considered to be a continuous variable, other methods like logistic regression or support vector machines do not work.

Through the described way of selecting variables, the best predictors were 4, 5, 6, 9, 10, 11, 12, 14, and 17. One needs to bear in mind that some of the linguistic variables are highly correlated among each other. Thus, those predictors with smaller intercorrelations were selected, which can have a lower correlation with the criterion. The obtained statistic model has an R^2 of .69, that means it can explain the variance in the students' estimation of text difficulty by about 70%. If one looks at the R^2 obtained in a 3-, 5-, or 10-fold cross-validation, the value gets lower, namely to .54, .50, and .38 respectively.

With these predictors, we get a formula for the readability of Latin literature (the sequence of predictors in the formula corresponds to their inclusion in the statistical model during stepwise linear regression):

$$\begin{aligned} f(x) = 14.478 + 24.885x_5 + 9.872x_{11} - 0.015x_4 \\ - 9.473x_{12} - 15.215x_{17} + 0.402x_{14} \\ - 0.097x_9 + 2.395x_{10} - 7.141x_6 \end{aligned}$$

3. Conclusions and Future Work

We have created a readability formula for Latin consisting of nine linguistic factors from various linguistic categories, which can explain the difficulty of Latin texts by about 70%, similar to other models (e.g., François and Fairon, 2012, have created a model with R^2 of .73). The formula presented in this paper could be further improved by adding more text to the corpus. In doing so, one could enhance the slightly lower R^2-values in cross-validation. A reason that those metrics are behind the model of François and Fairon (2012) could be due to the fact that Latin texts, unlike modern schoolbook texts, were composed for a highly educated upper class. All examined texts possess significant literary merit and are not merely instructional or exercise texts. Furthermore, there is the possibility of providing two separate formulas, one for prose and one for poetry texts.

Indeed, if one looks at the correlation between the difficulty of Latin poetry texts and certain linguistic variables, one can find some predictors with much higher correlation, e.g. the percentage of one syllable words correlates with r = -.50, the percentage of words outside a list of the most frequent 750 Latin words correlates with r = .60, and the number of interlaced hyperbata correlates with r = .55. A statistic model based only on poetic text could explain the variance in text difficulty of those text by 87%, but the prognostic power is much lower: one finds R^2 obtained in a 3-, or 5-fold cross-validation of .61, and .21, respectively.

Building upon the final readability model, further investigations can be conducted. By examining the residuals between the model and actual difficulty assessments, insights can be gained into which text genres and contents are generally easier or more challenging for readers to access. It can be expected that narrative passages are easier to understand than, for instance, philosophical treatises.

Since the formula provides a score for objective text difficulty that eliminates the personal characteristics of readers, in a concluding step, investigations can also be conducted on personal predictors. Especially in the context of education, it could be explored what personal prerequisites, particularly in vocabulary and grammar, one should have to understand a text.

4. Bibliographical References

Bayer, K. (2003). Bestimmung des Schwierigkeitsgrades von lateinischen Klassenarbeiten, *Pegasus*, 3(2):1–19.

Benjamin, R.G. (2012). Reconstructing Readability: Recent Developments and Recommendations in the Analysis of Text Difficulty, *Educational Psychology Review*, 24:63–88.

Berendes, K., Vajjala, S., Meurers, D., Bryant, D., Wagner, W., Chinkina, M., and Trautwein, U. (2018). Reading demands in secondary school: Does the linguistic complexity of textbooks increase with grade level and the academic orientation of the school track?, *Journal of Educational Psychology*, 110(4):518–543.

Burns, P.J. (2023). LatinCy: Synthetic Trained Pipelines for Latin NLP. *arXiv preprint arXiv:2305.04365.*

Dale, E. and Chall, J.S. (1948). A Formula for Predicting Readability, *Educational Research Bulletin*, 27:11–20+28.

Flesch, R. (1948). A New Readility Yardstick, *Journal of Applied Psychology*, 32:221–233.

François, Th. and Fairon, C. (2012). An “AI readability” formula for French as a foreign language. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 466–477, Jejudo, South Korea.

Friedrich, M. (2017). *Textverständlichkeit und ihre Messung. Entwicklung und Erprobung eines Fragebogens zur Textverständlichkeit.* Münster and New York.

Futrell, R., Mahowald, K., and Gibson, E. (2015). Large-scale evidence of dependency length minimization in 37 languages, *PNAS*, 112(33):10336–10341.

Gray, W.S. and Leary, B.E. (1935). *What Makes A Book Readable*. Chicago.

Groeben, N. (1982). *Leserpsychologie: Textverständnis – Textverständlichkeit*. Münster.

Gruber-Miller, J. and Mulligan, B. (2022). Latin Vocabulary Knowledge and the Readability of Latin Texts: A Preliminary Study, *New England Classical Journal*, 49(1):80–101.

Hancke, J., Vajjala, S., and Meurers, D. (2012). Readability Classification for German using lexical, syntactic, and morphological features. In *Proceedings of COLING 2012: Technical Papers*, pp. 1063–1080, Mumbai.

Haug, D. (2017). Syntactic discontinuities in Latin. A treebank-based study, *Bergen Language and Linguistics Studies*, 8:75–96.

Kintsch, W. (1988). The Role of Knowledge in Discourse Comprehension: A Construction-Integration Model. *Psychological Review*, 95(2):163–182.

Kintsch, W. and Miller, J.R. (1984). Readability: A View from Cognitive Psychology. In J. Flood (Ed.), *Understanding Reading Comprehension: Cognition, Language, and the Structure of Prose*. Newark, Del., pp. 220–232.

Kintsch, W. and Vipond, D. (1979). Reading Comprehension and Readability in Educational Practice and Psychological Theory. In L.-G. Nilsson (Ed.), *Perspectives on Memory Research: Essays in Honor of Uppsala University’s 500th Anniversary*. Hillsdale, NJ, pp. 329–365.

Kubát, M., Matlach, V., and Čech, R. (2014). *QUITA: Quantitative Index Text Analyzer*. Lüdenscheid.

McNamara, D.S., Graesser, A.C., McCarthy, P.M., and Cai, Z. (2014). *Automated Evaluation of Text and Discourse with Coh-Metrix*. New York.

Mikros, G. and Voskaki, R. (2021). A Modern Greek readability tool: Development of evaluation methods. In A. Pawłowski, J. Mačutek, Sh. Embleton, & G. Mikros (Eds.), *Language and Text: Data, models, information and applications*. Amsterdam and Philadelphia, pp. 163–175.

Rost, D.H. (2018). Leseverständnis. In D.H. Rost, J.R. Sparfeldt, & S.R. Buch (Eds.), *Handwörterbuch Pädagogische Psychologie*. Weinheim and Basel, 5th edition, pp. 494–506.

Selzer, J. (1981). Readability Is a Four-Letter Word, *Journal of Business Communication*,18:23–34.

Todirascu, A., François, Th., Gala, N., Fairon, C., Ligozat, A.-L., and Bernhard, D. (2013). Coherence and Cohesion for the Assessment of Text Readability. In *Proceedings of 10th International Workshop on Natural Language Processing and Cognitive Science (NLPCS 2013)*, pp. 11–19, Marseille, France.

Vajjala, S. (2022). Trends, Limitations and Open Challenges in Automatic Readability Assessment Research. In *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pp. 5366–5377, Marseille, France.

Vogel, M. and Washburne, C. (1928). An Objective Method of Determining Grade Placement of Children's Reading Material, *The Elementary School Journal*, 28(5):373–381.

Xia, M., Kochmar, E., and Briscoe, T. (2016). Text Readability Assessment for Second Language Learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 12–22, San Diego, Cal.

Zipf, G.K. (1935). *The Psycho-Biology of Language. An Introduction to Dynamic Philology*. Boston.

5. Language Resource References

Johnson, K.P., Burns, P.J., Stewart, J., and Todd, C. 2014–2021. *CLTK: The Classical Language Toolkit*. https://github.com/cltk/cltk.

Short, W.M. 2024. *Latin WordNet 2.0*. https://latinwordnet.exeter.ac.uk

Automatic Normalisation of Middle French and its Impact on Productivity

Raphael Rubino*, **Sandra Coram-Mekkey**†, **Johanna Gerlach***,
Jonathan Mutal*, **Pierrette Bouillon***
*TIM/FTI, University of Geneva, 1205 Geneva, Switzerland
{firstName.lastName}@unige.ch
†Fondation de l'Encyclopédie de Genève, Switzerland
coram.mekkey@gmail.com

Abstract

This paper presents a study on automatic normalisation of 16th century documents written in Middle French. These documents present a large variety of wordforms which require spelling normalisation to facilitate downstream linguistic and historical studies. We frame the normalisation process as a machine translation task starting with a strong baseline leveraging a pre-trained encoder–decoder model. We propose to improve this baseline by combining synthetic data generation methods and producing artificial training data, thus tackling the lack of parallel corpora relevant to our task. The evaluation of our approach is twofold, in addition to automatic metrics relying on gold references, we evaluate our models through post-editing of their outputs. This evaluation method directly measures the productivity gain brought by our models to experts conducting the normalisation task manually. Results show a $20+$ token per minute increase in productivity when using automatic normalisation compared to normalising text from scratch. The manually post-edited dataset resulting from our study is the first parallel corpus of normalised 16th century Middle French to be publicly released, along with the synthetic data and the automatic normalisation models used and trained in the presented work.

Keywords: Intralingual diachronic translation, Middle French, Archive, Normalisation, Productivity

1. Introduction

In Switzerland, each canton safeguards administrative, legal and financial documents produced by its successive governments. These large archives contain the oldest publicly released documents about the institutional history of Switzerland. A specific subset of these archives is the focus of our study, namely the Geneva Council Registers (in French: *les Registres du Conseil de Genève*), containing minutes of the council meetings covering local administrative and political decisions. These handwritten registers were held daily and are still being published nowadays as digital documents. They are an invaluable resource for studying the political, legal, economic, social, and religious history of the Geneva canton. More particularly during the 16th century, the Swiss Protestant Reformation took place. During this time, John Calvin played a major role in the Reformation and is considered today as one of the founders of Calvinism, a major branch of Protestantism (Backus and Benedict, 2011). Thus, Geneva Council Registers produced during this time period are interesting for historians, as the local political and religious decisions influenced the Geneva region and other Swiss cantons.

Nowadays, some of the original 16th century Geneva Council Registers (noted RCs hereafter) are available as digitised documents including a few with OCR. The registers produced between 1536 and 1544 entitled *Registres du Conseil de Genève à l'époque de Calvin* (Geneva Council Registers in Calvin's time) are available as hard copies. These archival documents were written in Middle French, a variant of the French language used mostly from the 14th to the 16th century (Buchi et al., 2019).[1] Furthermore, the textual content of these documents is sometimes mixed with Latin. These characteristics make RCs difficult to understand for non experts.

The current effort conducted by historians and palaeographers consists in editing the RCs textual content, mainly focusing on orthographic normalisation of various Middle French wordforms. This variety in spelling is due to the lack of language norms for Middle French during the 16th century, even for patronyms and toponyms. The resulting normalised textual content should follow editorial choices in terms of spelling normalisation and local grammatical modifications but should not contain syntactic alterations. One of the main motivations in normalising Middle French is to make RCs understandable to a wide audience. However, manual normalisation is a challenging and time consuming task.

In this study, we propose to assist the work currently conducted by historians and palaeographers in normalising the various wordforms observed in

[1] The exact time period when Middle French was spoken and written is still subject to debate among experts.

the RCs in the time of Calvin. Based on recent studies on spelling normalisation for French, we frame the task as a translation task (Bawden et al., 2022) in a very low resource setting. We leveraged pre-trained encoder–decoder large language models (LLMs), fine-tuned them with manually normalised RCs, to constitute a strong baseline. To improve over this baseline, we propose to enrich the available hand-crafted data with automatically generated parallel data, combining generative model prompting and back-translation (Marie and Fujita, 2021; Tonja et al., 2023). Our final model shows improved performances measured by automatic metrics compared to the baseline and to previously released normalisation models for French. Additionally, we present a qualitative analysis which highlights some of the differences between our approach and the baseline model.

To validate these findings, we conduct a manual evaluation to measure the post-editing time and effort spent by experts in correcting the automatically normalised RCs. The results show productivity gains in terms of normalisation throughput when using fine-tuned LLMs compared to manually normalising RCs from scratch. Moreover, our approach relying on synthetic data outperforms the fine-tuned LLMs making use of hand-crafted data only, both in terms of automatic metrics and productivity gain. To summarize, the contribution of our work is twofold: i) we describe a parallel data generation method which was not employed for historical text normalisation in previous work, and ii) we show that fine-tuning LLMs with a small amount of data greatly reduces the manual labour required to normalise Middle French, and can be further improved by using synthetic data.

The remainder of this paper is organised as follows. In Section 2, we introduce the background work for historical text normalisation, focusing on variants of the French language, before motivating our approach. In Section 3, the manual normalisation process is first presented, followed by the description and evaluation of the automatic normalisation process. The productivity gain expertiments based on post-editing is then detailed in Section 4 along with the corresponding results in terms of normalisation throughput. Finally, we conclude our study and present future work in Section 5.

2. Related Work

In this Section, we present the context of our normalisation work which is part of a larger project on Middle French modernisation. Then, previous work on spelling normalisation is introduced, followed by details about available resources relevant to our task. Finally, approaches to leveraging LLMs in low-resource scenario are described, before presenting the current limits of historical texts evaluation methods.

2.1. Context of the Study

The study presented in this paper and focusing on orthographic normalisation of RCs written in 16th century Middle French is part of a larger project which aims at producing a semantic and multilingual online edition of the Geneva Council Registers for the years 1545 to 1550. This project is based on a synergy between two faculties of the University of Geneva, the Centre universitaire d'informatique (CUI) and the Faculty of translation and interpreting (FTI), as well as the Fondation de l'Encyclopédie de Genève. The technical aspect of the project in terms of natural language processing is to automatise the normalisation and modernisation of RCs content, and to develop new functionalities that will make these archival documents accessible to a wide audience. Both normalisation and modernisation steps are leveraging low-resource machine translation techniques to process RCs, including fine-tuning large language models (LLMs) and producing artificial data as presented in this study. Each processing step applied to RCs will result in a version of the corpus, eventually resulting in multiple versions of RCs linked through token alignments. The assumption is that linked normalised and modernised RCs content will provide a useful source of knowledge for further research.

2.2. Normalising Spelling Variants

Intralingual diachronic translation aims at modifying textual content to match linguistic features of a time period, as orthographic, grammatical and syntactic features might evolve over time for a given language. A large body of work has been conducted on this task, in particular on normalising spelling variants observed in historical texts. Seminal studies on historical wordforms normalisation relied on distance and rule-based approaches (Hauser and Schulz, 2007; Rayson et al., 2007; Baron et al., 2009; Bollmann et al., 2011; Pettersson et al., 2013a; Bollmann et al., 2014). More recently, Machine Translation (MT) techniques were applied to intralingual diachronic translation, including statistical and neural models (Sánchez-Martínez et al., 2013; Pettersson et al., 2013b; Bollmann and Søgaard, 2016; Korchagina, 2017; Bollmann, 2018; Tang et al., 2018).

2.3. Available Resources

Among recent Neural MT (NMT) architectures, the Transformer (Vaswani et al., 2017) has been used for a variety of Natural Language Processing (NLP) tasks, including for automatic normalisation of Early Modern French (Bawden et al., 2022). The authors of this previous work have shown that Statistical MT (SMT) still outperforms the NMT architectures tested, namely LSTM (Hochreiter and Schmidhuber, 1997) and Transformer, with or without an additional lexicon-based post-processing step. The Transformer model used in their experiments was released and constitutes one of the few pre-trained resources available for spelling normalisation of French. We propose to evaluate their model on our task as a comparison point.

However, this publicly available model, called *ModFR*[2], was trained using the *FreEMnorm* corpus (Gabay and Gambette, 2022) which contains texts taken from French literature of the 17th century.[3] Our task involves the normalisation of Middle French, a language mainly used from the 14th to the 16th century (Buchi et al., 2019), which differs from previous study on French normalisation. While there is no clear consensus among philology and history experts about the beginning and the end dates of Middle French usage, the Dictionary of Middle French (Martin et al., 2020) covers the lexicon used from the year 1330 to 1500. This difference in time period between available resources for French and our task could hinder the straightforward application of previous approaches, but motivates us to leverage pre-trained LLMs to bootstrap our work.

2.4. Leveraging LLMs and Synthetic Data

Recent advances in NLP have been fuelled by the use of LLMs pre-trained on large amounts of data. However, to the best of our knowledge, only a few studies used pre-trained LLMs on tasks involving historical texts. For instance, Klamra et al. (2023) used a generative model to produce synthetic parallel data of archaic to modern Polish. This dataset was then used to fine-tune pre-trained encoder–decoder neural models to perform automatic modernisation of Polish. In our study, we propose to apply existing synthetic data generation techniques to build a parallel corpus. More precisely, inspired by Marie and Fujita (2021) and Tonja et al. (2023), a generative model is used to produce target data further back-translated into source data. In our study, the source data consists in non-normalised text while the target data consists in its normalised version. To the best of our knowledge, this is the first study on Middle French normalisation using fine-tuned LLMs.

2.5. Evaluation

If automatic metrics have been widely used to evaluate automatic normalisation models, the impact of such models on the productivity of experts conducting manual normalisation of historical texts has yet to be measured. In this study, we frame the comparison between normalising from scratch and editing automatically normalised text as a post-editing task. This allows us to perform manual evaluation of NMT-based normalisation models in terms of productivity gain, in addition to reporting results obtained with automatic metrics relying on gold references manually produced.

3. Normalisation of Middle French

This Section describes the normalisation process conducted to reduce spelling variants observed in Middle French contained in RCs written during the 16th century. First, the manual normalisation process is explained, presenting the orthographic modifications applied to the source text and defining the editorial choices made by the experts in terms of normalised wordforms. Second, the training and evaluation of automatic normalisation models are described, along with the synthetic data production method.

3.1. Manual Normalisation

The RCs consist in minutes of meetings held daily by Geneva canton council members. They contain political, administrative and judiciary decisions. They constitute a crucial resource for historical studies of the region for a given time period. The digitisation process of these manuscripts has been an ongoing effort, consisting mostly in scanning physical books. The results is a set of archived documents composed of RCs from 1408 to 1855 being publicly available online.[4] Recently, experts such as historians and palaeographers have been manually transcribing RCs. The work described in this paper is based on the manually transcribed version of RCs, which is the largest relevant dataset for our task.

More precisely, RCs from 1536 to 1550 were manually transcribed by historians and palaeographers. This task also involved slight modifications of the

[2] https://huggingface.co/rbawden/modern_french_normalisation

[3] https://github.com/FreEM-corpora

[4] https://ge.ch/arvaegconsult/ws/consaeg/public/FICHE/AEGSearch

Corpus	Segments	Tokens		Vocabulary		avg. tokens/segment	
		source	target	source	target	source	target
RCs	71.8k	2.7M	–	74.7k	–	37.4	–
RC_pe	2.5k	87.0k	–	7.0k	–	34.9	–
RC_para	2.3k	87.8k	82.4k	7.6k	5.7k	38.4	36.0
RC_synth	1.3M	195.2M	176.5M	0.47M	0.34M	147.2	133.0

Table 1: Number of segments, tokens and vocabulary entries (k for thousands, M for millions) for the transcribed RCs (noted *RCs*), the synthetic data created in our study (*RC_synth*), as well as the RCs subsets of original–normalised parallel text (*RC_para*) and original text used for the post-editing experiments (*RC_pe*). The corpora were normalised, lowercased and tokenised using the scripts released with the Moses toolkit (Koehn et al., 2007) prior to extracting data statistics.

Wordforms	Meaning
embossiou, enbosseu, entonnoir	a funnel
faulccry, faulxcry, foulcry, forcri	an alarm call
lause, lauze, loze, lose	a flat stone, a tile
maysoner, maisonner	to build
treul, true, trué, truez	a press
Toponyms	
Allemagne, Allamaignie, Allemagnyes, Allemaigne, etc.	Germany
Genève, Genefe, Genesve, Genevez, Genff, etc.	Geneva
Strasbourg, Estrabour, Estrapurg, Extrabourg, Strasburg, etc.	Strasburg

Figure 1: Examples of various wordforms encountered in the 16th century RCs, their normalised form is underlined, along with their meaning. Variants of toponyms are presented in the bottom part while general nouns and verbs are in the top part. The lists of toponym variants are truncated due to the large amount of wordforms observed.

textual content for increased readability by non-experts. The resulting corpus is a digital version of the RCs for the given time period covering 15 years. Furthermore, RCs from 1536 to 1544 were published as hard copy books. Both the digital and the hard copy versions of this corpus were not orthographically normalised and still contain a variety of wordforms, as illustrated in Figure 1. In addition to the manual transcription task, experts are currently conducting the manual orthographic normalisation of RCs content, starting from the transcription already done.

Due to the lack of spelling norms for Middle French during the 16th century, a large variety of wordforms were used compared to modern French. The manual normalisation consists in applying local orthographic and grammatical modifications to the original RCs content while leaving potentially archaic syntactic structures untouched. The normalisation guidelines defined by experts are described in Appendix A. This process differs from the historical text *modernisation* task, as it does not aim at transforming Middle French texts into their contemporary version. The objective is to reduce the spelling variations observed in RCs by selecting single wordforms. The latter are decided by experts conducting the manual normalisation task and follow editorial guidelines. We illustrate the normalisation process in Figure 2.

The main motivation behind conducting the orthographic normalisation of RCs is to improve the readability of texts difficult to understand while preserving the original structure. This will facilitate research in the historical, geographical and genealogical fields, among many others, by replacing various spelling variants with a single one. The orthographic normalisation will also serve as the basis for the syntactic normalisation of the text, which will in turn lead to its modernisation in current French. The latter two objectives are planned as future work but are out of scope of the presented study.

As a result of the manual normalisation, we currently have at our disposal a parallel set of RCs published over six months, one month per year from 1545 to 1550 (noted *RC_para*). This dataset is a subset of the non-normalised RCs manually transcribed from 1536 to 1550 (noted *RCs*). Details about these two corpora, along with the synthetic data described in Section 3.2 (noted *RC_synth*) and the RCs subset dedicated to post-editing (noted *RC_pe*), are presented in Table 1. Due to the small size of our hand-crafted parallel corpus, we will perform 5-fold cross-validation for all our automatic normalisation experiments presented in Section 3.2.

3.2. Automatic Normalisation

The aim of automatic normalisation is to assist historians and palaeographers in their task of manual normalisation and ultimately reduce their workload. We first propose to compare the performances of a publicly available pre-trained normalisation model for Early Modern French to

Le mardy 9e de octobre 1548 – L'on fasse respondre aut president de sadicte lectre
Le mardi 9e d'octobre 1548 – L'on fasse répondre au président de sadite lettre
Tuesday, October 9, 1548 – We answer to the president about his letter

(Les marchandz de Geneve) - Lesquieulx hont presenté une supplication par laquelle ilz prient
(Les marchands de Genève) - Lesquels ont présenté une supplication par laquelle ils prient
(The merchants of Geneva) - Who have presented a supplication by which they pray

Et dempuys a esté resoluz qui soyt liberé publiquement, à voex de trompe, et aut tribunal ordinayre.
Et depuis a été résolu qui soit libéré publiquement à voix de trompe et au tribunal ordinaire.
And it has since been resolved that he be released publicly and in ordinary court.

Ledictz jour, vendredy 28 octobrix 1547, en l'Evesché
Ledit jour vendredi 28 octobris 1547 en l'Évêché
Said day Friday October 28 1547 in the bishop's house[a]

Ayme Richard, habitant et ferratier, filz de feu Thivent Richard, de Sonzier
Aimé Richard habitant et ferratier fils de feu Thivent Richard de Scionzier
Aimé Richard inhabitant and ironworker son of the late Thivent Richard of Scionzier

[a] The bishop's house, translation of *Evesché* in this example, refers to the house inhabited by the previous bishop which was converted into a prison.

Figure 2: Segments sampled from the RCs original–normalised parallel corpus in Middle French, with segments in their original form (top, colored), their normalised version (middle, in black, normalised words underlined) and a possible English translation (bottom, *italic*).

an out-of-the-box pre-trained LLM. We then make use of our parallel data (*RC_para*) consisting of manually transcribed RCs as source and their normalised version as target.

3.2.1. LLM Setup

Our preliminary experiments showed that *m2m100* (Fan et al., 2021) outperforms other pre-trained MT models when fine-tuned with our data. Thus, we decided to conduct all our experiments using this model in its *base* version (418M parameters). We used the publicly released checkpoint available with the HuggingFace Transformers library (Wolf et al., 2019).[5] The fine-tuned version of this model using our parallel data is the *baseline* in our study. The fine-tuning procedures employed in our experiments are detailed for all models in Appendix B.

3.2.2. Synthetic Data

Due to the lack of parallel data relevant to the RCs and written in Middle French, we generated synthetic parallel data with a two-step process: 1) generative model prompting for target data generation, followed by 2) normalised-to-non-normalised back-translation to obtain a parallel corpus (Marie and Fujita, 2021; Tonja et al., 2023). The generative model used was *Bloomz* with 560M parameters (Muennighoff et al., 2022). This model was fine-tuned with the target side of our parallel corpus written in normalised Middle French. As this fine-tuning step relies on the training data taken from *RC_para*, it was conducted individually for each of the 5 folds. The motivation behind fine-tuning the generative model is to increase the relevancy of automatically generated data for the task at hand. Once the fine-tuning step was done, we proceed with prompting the model to produce synthetic data. The prompting method consisted in inputting sequences composed of consecutive tokens taken from the target side of *RC_para*, the same corpus used to fine-tune the generative model.[6]

The resulting target-side corpus automatically generated was then back-translated into the non-normalised source side of the synthetic parallel corpus. The back-translation model was trained on the combination of the *RC_para* corpus with the automatic translation of the *RCs* corpus.[7] The resulting parallel corpus, presented in Table 1 and noted *RC_synth*, was used to perform continued training of the pre-trained LLM (model noted *synthetic*) (Gururangan et al., 2020).[8] The average

[5] `https://huggingface.co/facebook/m2m100_418M`

[6] We used between 8 and 12 tokens as prompts to obtain different results and combine all the generated data.

[7] The automatic translation of the *RCs* corpus was obtained using the *baseline* model.

[8] A few samples of the produced synthetic data are presented in Appendix C.

model	BLEU	chrF	TER	WER	acc.
identity	24.2	65.8	45.0	42.5	13.4
m2m100	23.1	57.5	54.1	66.0	1.5
ModFR	32.3	71.1	38.5	38.8	11.9
baseline	79.7	91.1	11.9	6.7	47.4
synthetic	81.8*	92.2*	11.4	11.6	36.2
synthetic+ft	83.5*	93.6*	9.0*	5.7	47.8

Table 2: Averaged test results (5-fold cross-validation) measured by automatic metrics for the orthographic normalisation task of RCs, comparing the identity function (copy of the source) to previously released models (top part) and to our approach (bottom part). For BLEU, chrF and segment-level accuracy (*acc.*), the higher the better, while the lower the better for TER and WER. Results marked with * are significantly better than previous rows with $p < 0.01$, based on the paired bootstrap resampling technique with 1000 resamples.

number of tokens per segment for *RC_synth* is larger than for *RCs* and *RC_para* because we do not truncate the generated sequences, but instead let the generative model produce the end of sequence token. Finally, we fine-tune the resulting model using RCs parallel corpus (model noted *synthetic+ft*).

3.2.3. Automatic Metrics

The automatic evaluation was conducted using popular MT metrics, namely BLEU (Papineni et al., 2002), chrF (Popović, 2015) and TER (Snover et al., 2006), implemented in the SacreBLEU toolkit (Post, 2018).[9] For these three metrics, significance testing using paired bootstrap resampling with 1000 resamples was conducted to compare the *baseline*, *synthetic* and *synthetic+ft* models (Koehn, 2004). In addition, we measured the word error rate (WER) and the segment-level accuracy reached by the evaluated models. We believe that the latter metrics allow to grasp the manual effort required to produce publishable normalised text.

3.2.4. Quantitative Analysis

The 5-fold cross-validation test results measured by automatic metrics are presented in Table 2. We averaged results over the 5 runs, each run consisting in 60% of *RC_para* used as training set, 20% as validation and 20% as test (roughly 1.4k, 450 and 450 segments for the train, validation and test sets respectively). We evaluated a previously released normalisation model for Early Modern French (noted *ModFR*) (Bawden et al., 2022), along with a non-fine-tuned pre-trained LLM (*m2m100*). As an additional comparison point, we also considered the identity function, i.e. leaving the source non-normalised text untouched and comparing it to the normalised reference (noted *identity*). Finally, three fine-tuned versions of the *m2m100* model were also evaluated, namely the *baseline* model which was fine-tuned using the *RC_para* corpus only, the *synthetic* and *synthetic+ft* models which were fine-tuned using the *RC_synth* and *RC_synth+RC_para* respectively. The latter model was trained following a two-step process: continued training with *RC_synth* followed by fine-tuning with *RC_para*.

The results obtained with the segment-level automatic metric (*acc.*) show that previously released models do not outperform the identity function. The three MT-oriented metrics, namely BLEU, chrF and TER, as well as WER, show that *ModFR* outperforms both the identity function and out-of-the-box *m2m100*. Both the baseline and our final model (*synthetic+ft*) outperform the previously released model for Early Modern French according to the five metrics used. Adding synthetic data to the hand-crafted parallel corpus improves normalisation performances at the n-gram (BLEU), token (TER) and character (chrF) levels. However, when using synthetic data only without the final fine-tuning (model noted *synthetic*), a 11.2pts drop in terms of segment-level accuracy is observed, while gains are observed with MT metrics. This indicates that synthetic data improves normalisation at the n-gram, token and character levels, but introduce errors which lower the number of correctly normalised full segments. Finally, we see that our final model reaches the best scores overall, validating our synthetic data generation approach and confirming the need to eventually fine-tune the model using hand-crafted parallel data.

3.2.5. Coverage Analysis

As an additional experiment to help analyse the automatic normalisation results, we computed the rates of source-side out-of-vocabulary (OOV) tokens between the test set of each fold and the training sets used in our experiments, namely *RC_para* and *RC_synth*. We also included the *FreEMnorm* (Gabay and Gambette, 2022) corpus in the OOV rates calculation as it was used to train the *ModFR* (Bawden et al., 2022) model. We lowercased and tokenised all datasets prior to computing these rates, using the scripts released with the *Moses* toolkit (Koehn et al., 2007). We present the OOV results in Figure 3 for each fold in

[9] SacreBLEU signatures: version:2.3.1|nrefs:1
case:mixed|eff:no|tok:13a|smooth:exp
case:mixed|eff:yes|nc:6|nw:0|space:no
case:lc|tok:tercom|norm:no|punct:yes|asian:no

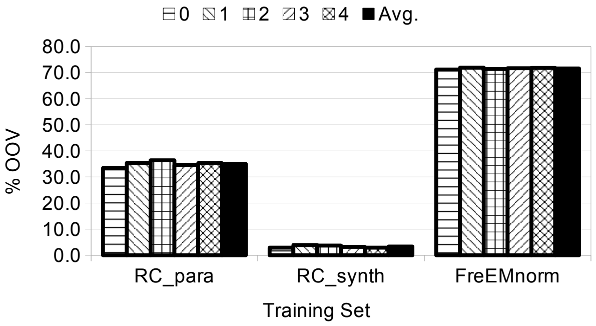


Figure 3: Out-of-vocabulary rates (%) for test tokens wrt. the training sets used in our experiments and the *FreEMnorm* (Gabay and Gambette, 2022) corpus. Hatched bars represent each fold individually (from 0 to 4) and the solid bar represents the averaged rate over 5 folds.

order to show that no particular fold suffered from a lower token-level coverage compared to the other folds.

The OOV rates clearly indicate that the *FreEMnorm* corpus provides a lower vocabulary coverage compared to the *RC_para* training set (71.6% vs. 35.0% OOV rates respectively). The low coverage of the Early Modern French corpus could partially explain the normalisation performances reached by the *ModFR* model on our Middle French data. Surprisingly, the source side of the synthetic data (resulting from the back-translation of the generative model output) reaches an average OOV rate of 3.3%, a 31.7pts absolute decrease compared to the average OOV rate obtained with the *RC_para* training sets. This particular result validates the use of synthetic data for vocabulary coverage in a low-resource scenario. However, while synthetic data is relatively cheap to produce, this approach still requires a small amount of well-formed target data to fine-tune the generative model.

3.2.6. Qualitative Analysis

To assess the strengths and weaknesses of the *baseline* and *synthetic+ft* models on specific elements to be normalised, we conduct a qualitative analysis of the automatically normalised segments. While the *baseline* model reaches relatively high performances compared to the other models, the *synthetic+ft* model is better at normalising the spelling of proper nouns and verbs, as presented by the examples in Appendix D. In the first example, the spelling of the proper noun *Pregnier* in the source segment should be normalised as *Pregny* but the *baseline* failed to do so while the *synthetic+ft* normalised it correctly. Similarly, in the third example, *Dolle* is normalised as *Dole* with the model using synthetic data. In terms of verb spelling, in the second example, *Doygbe* is correctly normalised as *Doive* by *synthetic+ft*, and in the third example, *requesté* is normalised as *requêté*.

Both models, however, introduce errors for source tokens which should not be modified, i.e. when no normalisation is necessary according to the gold reference. For instance, in Appendix D, the second example shows that the verb *levés* is correctly spelled in the source and reference segments while both the *baseline* and *synthetic+ft* models remove the plural form and rewrite it as *levé*. Overall, at the segment-level according to the *chrF* metric on the validation set, *synthetic+ft* is better than *baseline* for approx. 30% of the segments and both models are equal for approx. 55% of the segments. These results show that for approx. 15% of the segments, *baseline* is better than *synthetic+ft*.

4. Post-editing and Productivity Gain

One of the aims of this study is to measure the productivity gain achieved by using automatic normalisation followed by post-editing, compared to manually normalising from scratch. Moreover, we would like to validate the results obtained with automatic metrics in our previous experiments (cf. Table 2). Our post-editing experiments make use of a subset taken from the non-normalised source corpus, covering 5 months of the year 1545, which consists in approx. 2500 segments. Details about the dataset used are presented in Table 1 and noted *RC_pe*. The segments contained in *RC_pe* were normalised by our systems, namely *baseline* and *synthetic+ft*, or kept as is (i.e. the identity function), before being randomly presented to a human expert for post-editing.[10] We removed target segments which were identical between the two normalisation models and the identity function (approx. 500 segments were removed). The post-editing platform used in our experiments is *COPECO* (Mutal et al., 2020).

To conduct the post-editing task, we relied on a single historian who is an expert in 16th century Middle French texts and has participated in the manual transcriptions of RCs. The time spent on each segment, as well as the number of keystrokes for each segment, were measured during the post editing task. Due to the difficulty of this task even for trained experts, the set of segments to be post-edited was split in subtasks of approx. 100 segments. In order to limit the impact of normalising short and long segments on the final results, we kept segments containing between 2 and 128

[10] The same post-editing platform was used to post-edit all segments, including the source segments in case of the identity function and the automatically normalised segments as well. The post-editing interface is presented in Appendix E.

model	segments	tokens	keystrokes/token	time/token (sec)	token/minute	segment/hour
identity	385	12.1k	1.86	2.55	23.5	45.0
baseline	833	27.1k	0.42	1.33	45.3	83.4
synthetic+ft	834	28.1k	0.36	1.19	50.5	90.0

Table 3: Manual post-editing of RCs, comparing the identity function (copy of the source) to our approach (*baseline* and *synthetic+ft*) in terms of number of keystrokes per token, the time in seconds spent per token, the number of tokens processed per minute and the number of segments processed per hour. A larger number of segments were post-edited for the *baseline* and *synthetic+ft* systems compared to *identity* as we noticed a smaller gap in productivity gains between the outputs coming from the two former models.

model	BLEU	chrF	HTER	WER	acc.
identity	21.2	63.3	47.7	50.7	0.8
baseline	79.5	91.3	10.1	9.5	31.5
synthetic+ft	84.9	94.3	6.5	7.9	36.5

Table 4: Automatic metrics scores obtained when evaluating automatically normalised outputs using their manually post-edited version as reference, comparing the identity function (copy of the source) to our approach. For BLEU, chrF and segment-level accuracy (*acc.*), the higher the better, while the lower the better for HTER and WER.

tokens. Furthermore, we removed segments for which the post-editing time exceeded 5 minutes. Finally, segments for which 0 keystrokes were recorded but with a post-editing time exceeding 0.5 seconds were removed.

The results obtained in terms of normalisation productivity are presented in Table 3. The post-editing results show that both the *baseline* and the *synthetic+ft* models lead to increased normalisation productivity compared to normalising RCs from scratch. This is clearly shown by an increase in normalised token per minute and segment per hour. The number of keystrokes per token decreases with automatic normalisation compared to fully manual normalisation. Between the *baseline* and the *synthetic+ft* models, we observe a processed token per minute rate of 45.3 and 50.5 respectively. When measuring the number of segments processed per hour, an increase of 6.6 segments is reached by the model using synthetic data compared to the baseline.

These findings corroborate the results obtained in Table 2 with automatic metrics. We conducted further evaluations with the latter metrics to measure the distance between the models' outputs and their manually post-edited version. The results in terms of automatic metrics using the post-edited target as gold reference are presented in Table 4. We observe with these results that our final model (*synthetic+ft*) outperforms the baseline by 5.4pts BLEU and 5.0pts segment-level accuracy. Comparing results presented in Table 4 to results presented in Table 2, we see a decrease in performances when normalising the *RC_pe* corpus compared to normalising the *RC_para* corpus. This could be due to the lack of vocabulary coverage in the RCs from 1545, as *RC_para* is a mix of RCs covering one month per year from 1545 to 1550. We noticed that the RCs content vary from one year to another in terms of vocabulary, which is due to the various topics of discussion changing over time.

5. Conclusion

This paper presented a study on 16th century Middle French spelling normalisation. We compiled a dataset taken from the publicly available Geneva Council Registers which were manually transcribed, before manually normalising a subset of this corpus to build a parallel normalisation corpus. A strong baseline based on a pre-trained LLM, fine-tuned on the hand-crafted parallel corpus, was shown to outperform a previously released model trained for the normalisation of Early Modern French, as indicated by automatic metrics. Further experiments with synthetic data generation improved over this baseline at the segment, n-gram, token and character levels.

To validate these findings, we conducted a manual evaluation based on a post-editing task, comparing normalisation from scratch to the proposed approach. We show that fine-tuning a multilingual pre-trained LLM with a small amount of normalised parallel data increases the productivity of human experts by a relative gain of 92.8% in terms of normalised tokens per minute, compared to manually normalising text from scratch. Furthermore, adding synthetic data to the LLM fine-tuning increases productivity compared to the baseline by 5.2 tokens per minute, a 114.9% gain relative to full manual normalisation. It is, to the best of our knowledge, the first study on productivity gain measured through post-editing of 16th century Middle French archival documents normalisation.

As future work, we plan to run our approach iteratively, making use of the manually post-edited data to improve the performances of our automatic normalisation model. The next step in the ongoing Middle French modernisation project is to conduct normalisation at the syntactic level, in addition to the current local orthographic and grammatical normalisation. In addition, we will explore various prompting techniques in order to obtain more relevant synthetic data from generative models. Finally, due to the change in topics discussed during Council meetings depending on the local events, we will conduct a diachronic study, measuring the impact of using temporally-related training and test data, compared to randomly sampling segments from the whole RCs content as we did in this study.

Limitations

We recognize the following limitations of this work.

First, the experiments were conducted on a variant of the Middle French language from the 16th century. Middle French has evolved over time and our work is considering a relatively narrow time frame in the history of this language.

Second, only a few pre-trained language models were tested during our preliminary experiments relatively to the large number of models currently publicly available. Some of these models were pre-trained on Modern or Early Modern French language, while other models were trained jointly on several languages, including languages relevant to our work such as Latin. Therefore, the models selected in our study may not be representative of all publicly released pre-trained models in terms of languages, number of parameters, training objectives nor architectures.

Third, the hand-crafted corpus produced in our work is relatively small in terms of number of tokens and vocabulary size compared to commonly used corpora in natural language processing experiments. This is mainly due to the high cost of producing such dataset for which the expertise of historians and palaeographers is required, while following strict editorial guidelines.

Finally, the post-editing experiments conducted in our work involves a single human expert. This is due to the nature of the task itself, requiring strong expertise in 16th century history, geographical knowledge of the Geneva canton, as well as a solid philological background to allow for Middle French normalisation and local grammatical alterations.

Ethical Considerations

The dataset hand-crafted in our study is based on publicly available archives from the 16th century (non-license, public domain). We reviewed the content of the documents selected for manual normalisation and we believe that this resource represents accurate historical events. However, some textual elements of this corpus could be considered as toxic and harmful, or disrespectful of the privacy of the people and places mentioned in these archives. We thus made sure that all data used in our work and to be released as part of our parallel datasets are in the public domain and already freely available. Consequently, no increased risks or harm is caused by our dataset. Instead, it serves as a resource for historical studies and digital humanities.

The fine-tuned models to be released with our work are based on publicly released and licensed pre-trained models (MIT License). We respect the permissions to use, modify and distribute the models. We will release the fine-tuned models under the MIT License.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments.

This work was partially funded by the FNS (Fonds national suisse), SSH 2022 grant n. 215733, for the project entitled *Une édition sémantique et multilingue en ligne des registres du Conseil de Genève (1545-1550)*, acronym *RCnum*.

All experiments were conducted on the University of Geneva computing cluster HPC *Baobab* and *Yggdrasil*.

Bibliographical References

Irena Backus and Philip Benedict. 2011. *Calvin and his influence, 1509-2009*. Oxford University Press.

Alistair Baron, Paul Rayson, and DE Archer. 2009. Automatic standardization of spelling for historical text mining. *Proceedings of Digital Humanities 2009*.

Rachel Bawden, Jonathan Poinhos, Eleni Kogkitsidou, Philippe Gambette, Benoît Sagot, and Simon Gabay. 2022. Automatic normalisation of early modern french. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3354–3366.

Marcel Bollmann. 2018. *Normalization of historical texts with neural network models*. Ph.D. thesis, Dissertation, Bochum, Ruhr-Universität Bochum, 2018.

Marcel Bollmann, Florian Petran, and Stefanie Dipper. 2011. Rule-based normalization of historical texts. In *Proceedings of the Workshop* *on Language Technologies for Digital Humanities and Cultural Heritage*, pages 34–42.

Marcel Bollmann, Florian Petran, and Stefanie Dipper. 2014. Applying rule-based normalization to different types of historical texts—an evaluation. In *Human Language Technology Challenges for Computer Science and Linguistics: 5th Language and Technology Conference, LTC 2011, Poznań, Poland, November 25–27, 2011, Revised Selected Papers 5*, pages 166–177. Springer.

Marcel Bollmann and Anders Søgaard. 2016. Improving historical spelling normalization with bi-directional lstms and multi-task learning. *arXiv preprint arXiv:1610.07844*.

Eva Buchi, Bernard Combettes, Veronika Lux-Pogodalla, Béatrice Stumpf, Gilles Toubiana, and Delphine Barbier-Jacquemin. 2019. Histoire du français–frise chronologique.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Simon Gabay and Philippe Gambette. 2022. FreEM-corpora/FreEMnorm: FreEM norm Parallel (original vs. normalised) corpus for Early Modern French.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Andreas W Hauser and Klaus U Schulz. 2007. Unsupervised learning of edit distance weights for retrieving historical spelling variations. In *Proceedings of the first workshop on finite-state techniques and approximate search*, pages 1–6.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Cezary Klamra, Katarzyna Kryńska, and Maciej Ogrodniczuk. 2023. Evaluating the use of generative llms for intralingual diachronic translation of middle-polish texts into contemporary polish. In *International Conference on Asian Digital Libraries*, pages 18–27. Springer.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180. Association for Computational Linguistics.

Natalia Korchagina. 2017. Normalizing medieval german texts: from rules to deep learning. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pages 12–17.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Benjamin Marie and Atsushi Fujita. 2021. Synthesizing monolingual data for neural machine translation. *arXiv preprint arXiv:2101.12462*.

Robert Martin, Sylvie Bazin, and Pierre Cromer. 2020. Dictionnaire du moyen français. *ATILF-CNRS & Université de Lorraine*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Jonathan Mutal, Pierrette Bouillon, Perrine Schumacher, and Johanna Gerlach. 2020. Copeco: a collaborative post-editing corpus in pedagogical context. In *Proceedings of 1st Workshop on Post-Editing in Modern-Day Translation*, pages 61–78.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Eva Pettersson, Beáta Megyesi, and Joakim Nivre. 2013a. Normalisation of historical text using context-sensitive weighted Levenshtein distance and compound splitting. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, pages 163–179, Oslo, Norway. Linköping University Electronic Press, Sweden.

Eva Pettersson, Beáta Megyesi, and Jörg Tiedemann. 2013b. An smt approach to automatic annotation of historical text. In *Proceedings of the Workshop on Computational Historical Linguistics at NODALIDA 2013, NEALT Proceedings Series*, volume 18, pages 54–69.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Paul Rayson, Dawn Archer, Alistair Baron, and Nicholas Smith. 2007. Tagging historical corpora-the problem of spelling variation. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum fr Informatik.

Felipe Sánchez-Martínez, Isabel Martínez-Sempere, Xavier Ivars-Ribes, and Rafael C Carrasco. 2013. An open diachronic corpus of historical spanish: annotation criteria and automatic modernisation of spelling. *arXiv preprint arXiv:1306.3692*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

Gongbo Tang, Fabienne Cap, Eva Pettersson, and Joakim Nivre. 2018. An evaluation of neural machine translation models on historical spelling normalization. *arXiv preprint arXiv:1806.05210*.

Atnafu Lambebo Tonja, Olga Kolesnikova, Alexander Gelbukh, and Grigori Sidorov. 2023. Low-resource neural machine translation improvement using source-side monolingual data. *Applied Sciences*, 13(2):1201.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

A. Appendix: Normalisation Guidelines

The normalisation guidelines were defined by the historian in charge of manually normalising RC content. This person is an expert in 16th century Middle French, in the Geneva region and in the political landscape in Calvin's time. The same expert was in charge of post-editing the automatically normalised content produced by our models. The same guidelines were used when manually normalising RC content from scratch and when post-editing our models' output.

The normalisation applied to the source textual content is focused on local orthographic and grammatical elements while leaving syntactic structures unchanged. This normalisation process is part of a larger normalisation and modernisation effort, as well as lexical enrichment and indexing, as described in Section 2.1. The normalisation guidelines were the following:

- First characters are uppercased at the start of sentences, but also for patroyms and toponyms.
- Limit the use of ponctuation marks:
  - semicolons in lemmas only to separate different items,
  - commas before decisions, e.g. (regarding) ordered/stopped/solved,
  - periods at the end of sentences.
- Use of diacritical marks (apostrophes) except for cases where *que*, followed by a vowel, actually stands for *qui*, e.g. *sont survenues quelques lettres que attouchaient à Genève* (in English: a few letters about Geneva appeared).
- Extended emphasis and accentuation based on modern usage
- Gender and number of past participle agreement, e.g. *de celui qui les a baillé* becomes *de celui qui les a <u>baillés</u>*, *sus la supplication qui a présenté* becomes *sus la supplication qui a présentée*, except when there is a doubt such

as *lui soit baillé trois écus* **not** to be corrected in *lui soient baillés trois écus* because it is an ambiguous case: *trois écus* could be the object or the subject.

- Verb agreement, e.g. *ordonné que lesdits six écus lui soit délivrés* becomes *ordonné que lesdits six écus lui soient délivrés* (in English: ordered that the said six écus be delivered to him)
- Modernisation of patronyms, first names and toponyms.
- Correction of genders according to modern usage, e.g. *la dimanche* (in English: the Sunday) becomes *le dimanche*, *la reste* (in English: the rest) becomes *le reste*.
- Singular feminine possessive determiner replacement, e.g. *ma* (my), *ta* (your), *sa* (his, her, their), for nouns starting with a vowel or with a silent *h*, by the masculine forms *mon*, *ton*, *son*. For instance, *à sa humble requête* becomes *à son humble requête* (in English: to his/her/their humble request).

B. Appendix: LLM Fine-tuning Procedure

All models fine-tuned and evaluated in this work relied on the HuggingFace Transformers library (Wolf et al., 2019) with the Pytorch backend (Paszke et al., 2019). Models fine-tuning were conducted on single Nvidia RTX A5000 and 3090 GPUs with 24GB memory during a maximum of 100k steps (maximum of 12h) with early stopping if convergence is reached. We used batch sizes between 4 and 16 segments depending on training and testing phases. The optimizer used was AdamW (Loshchilov and Hutter, 2017), measuring BLEU scores on the validation set every 500 steps for the *baseline* and *synthetic+ft* models, and every 5000 steps for the *synthetic* model. The back-translation and normalisation models based on *m2m100* with 418M parameters were using the configuration released with the checkpoint, except for the learning rate. For the latter hyper-parameter, we searched for the best learning rate in a given range by monitoring performances obtained on the validation set. The learning rate search ranges were:

- *baseline* model: between $1e^{-6}$ and $2e^{-5}$
- *synthetic* model: between $8e^{-7}$ and $2e^{-5}$
- *synthetic+ft* model: between $8e^{-7}$ and $2e^{-6}$

The generative models were fine-tuned for 100k steps with a batch size of 4 using the AdamW optimizer. Three learning rates were used leading to three fine-tuned models: $8e^{-7}$, $1e^{-6}$ and $5e^{-6}$. The resulting models were finally averaged to compose the final generative model used to produce synthetic target data through prompting.

C. Appendix: Synthetic Parallel Data

The segments below are sampled from the *RC_synth* corpus, with the target side (in black, with differences underlined) produced by prompting a fine-tuned generative model before being back-translated to produce the source side (non-normalised, colored).

Accord passé entre Jehan Cuvat, ancien admodiataire du revenuz de l'Hospital, et François Beguin, consellier des comptes
Accord passé entre Jean Cuvat ancien amodiataire du revenu de l'hôpital et François Béguin conseiller des comptes

M. Morel, le tressorier Corne, disant qui ont remercié Dieu et la Ville de ne fere poyé aulchongs droys ny aulcunes retenues de ce qui a esté adjugé à l'Hospitall.
M. Morel le trésorier Corne disant qui ont remercié Dieu et la ville de ne faire payer aucuns droits ni aucunes retenues de ce qui a été adjugé à l'hôpital.

Deviser et conferir ensemble que ilz puissent aussi avoir conseilz de ceulx qui serontz expers.
Deviser et conférer ensemble qu'ils puissent aussi avoir conseil de ceux qui seront expers.

(Le seigneur Curteti, de Jussier) - Lequel a prier luy faire aulmone de ce que possede et des biens qui sera expirer, et l'a faict poyer.
(Le seigneur Curtet de Jussy) - Lequel a prié lui faire aumône de ce que possède et des biens qui sera expiré et l'a fait payer.

Et sur ce, ordonné qui soit faict ung prisonnier et que le chastellain se doibge enquerré de la verité du faict, et sus luy l'on fera justice.
Et sur ce ordonné qui soit fait un prisonnier et que le châtelain se doive enquerre de la vérité du fait et sur lui l'on fera justice.

Leur a esté par cy-devant imposé. Sur quoy, Messieurs du Petit Conseyl, il ont refferuz que hier, il furent informés que le seigneur Amyed Perrin, jadix ministre de Loys Bernard, lequelt avoyt malle servente avecque Claude Du Pan, lequell ont palliarder et ce que il avient fayct, ce ont estés chastiés, et maentenant il en ont

pour leur responces . . .
Leur a été par ci-devant imposée sur quoi messieurs du petit conseil ils ont référé que hier ils furent informés que le seigneur Ami Perrin jadis ministre de Louis Bernard lequel avait maille servante avec Claude Dupan lequel ont paillardé et ce que ils avaient fait ce ont été châtiés et maintenant ils en ont pour leurs réponses . . .

D. Appendix: Qualitative Analysis

The segments below are extracted from the validation set where the *synthetic+ft* model outperforms the *baseline* on verbs and proper nouns spelling.Underlined tokens are correctly normalised by *synthetic+ft* and erroneous with *baseline*.

source
(Les admodiataires et dismier de Pregnier; Michiel Mallet) - Lequel a requis qui plaise à Messieurs avoir regard sus la tempeste tombee sur leurs diesme etc.
baseline
(Les amodiataires et dîmeurs de Périgny Michel Malet) - Lequel a requis qui plaise à messieurs avoir regard sur la tempête tombe sur leur dîme etc.
synthetic+ft
(Les amodiataires et dîmeurs de Pregny Michel Malet) - Lequel a requis qui plaise à messieurs avoir regard sur la tempête tombée sur leur dîme etc.
reference
(Les amodiataires et dîmeur de Pregny Michel Maillet) - Lequel a requis qui plaise à messieurs avoir regard sur la tempête tombée sur leur dîme etc.
translation
(The lessees and the tithe stewards of Pregny Michel Maillet) - Who requested that the councillors consider the storm that fell on their tithe etc.

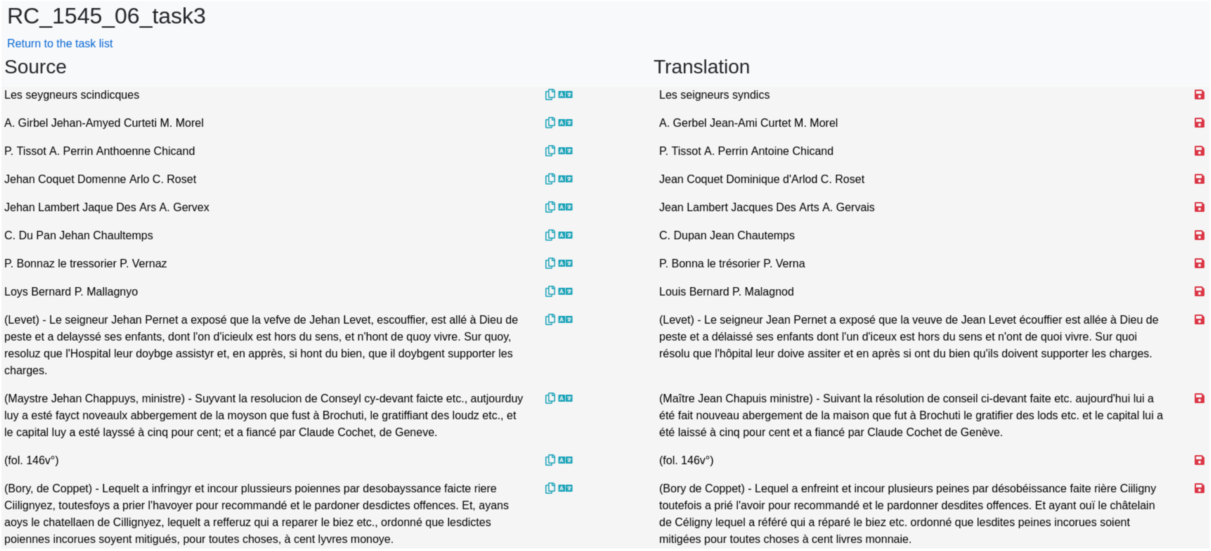
---

source
(Jacque-Nycolas Vulliet) - Doybge rendre les gages levés à cause qui ne cria pas.
baseline
(Jacques-Nicolas Vulliet) - Doyge rendre les gages levé à cause qui ne criera pas.
synthetic+ft
(Jacques-Nicolas Vulliet) - Doive rendre les gages levé à cause qui ne cria pas.
reference
(Jacques-Nicolas Vulliet) - Doive rendre les gages levés à cause qui ne cria pas.
translation
(Jacques-Nicolas Vulliet) - Must return the guarantees because he did not auction.

---

source
(La Guygona) - Laquelle a requesté luy oultroyer une lectre de faveur, affin avoir pour son mary detenuz en prison à Dolle etc.
baseline
(La Guygona) - Laquelle a requéré lui octroyer une lettre de faveur afin avoir pour son mari détenu en prison à Dolle etc.
synthetic+ft
(La Guyonay) - Laquelle a requêté lui octroyer une lettre de faveur afin avoir pour son mari détenu en prison à Dole etc.
reference
(La Guigone) - Laquelle a requêté lui octroyer une lettre de faveur afin avoir pour son mari détenu en prison à Dole etc.
translation
(La Guigone) - Who requested to grant her a letter of favor in order to have for her husband detained in prison in Dole etc.

E. Appendix: Post-editing Interface



RC_1545_06_task3

Return to the task list

Source	Translation
Les seygneurs scindicques	Les seigneurs syndics
A. Girbel Jehan-Amyed Curteti M. Morel	A. Gerbel Jean-Ami Curtet M. Morel
P. Tissot A. Perrin Anthoenne Chicand	P. Tissot A. Perrin Antoine Chicand
Jehan Coquet Domenne Arlo C. Roset	Jean Coquet Dominique d'Arlod C. Roset
Jehan Lambert Jaque Des Ars A. Gervex	Jean Lambert Jacques Des Arts A. Gervais
C. Du Pan Jehan Chaultemps	C. Dupan Jean Chautemps
P. Bonnaz le tressorier P. Vernaz	P. Bonna le trésorier P. Verna
Loys Bernard P. Mallagnyo	Louis Bernard P. Malagnod
(Levet) - Le seigneur Jehan Pernet a exposé que la vefve de Jehan Levet, escouffier, est allé à Dieu de peste et a delayssé ses enfants, dont l'on d'icieulx est hors du sens, et n'hont de quoy vivre. Sur quoy, resoluz que l'Hospital leur doybge assistyr et, en apprès, si hont du bien, que il doybgent supporter les charges.	(Levet) - Le seigneur Jean Pernet a exposé que la veuve de Jean Levet écouffier est allée à Dieu de peste et a délaissé ses enfants dont l'un d'iceux est hors du sens et n'ont de quoi vivre. Sur quoi résolu que l'hôpital leur doive assiter et en après si ont du bien qu'ils doivent supporter les charges.
(Maystre Jehan Chappuys, ministre) - Suyvant la resolucion de Conseyl cy-devant faicte etc., autjourduy luy a esté fayct noveaulx abbergement de la moyson que fust à Brochuti, le gratiffiant des loudz etc., et le capital luy a esté layssé à cinq pour cent; et a fiancé par Claude Cochet, de Geneve.	(Maître Jean Chapuis ministre) - Suivant la résolution de conseil ci-devant faite etc. aujourd'hui lui a été fait nouveau abergement de la maison que fut à Brochuti le gratifier des lods etc. et le capital lui a été laissé à cinq pour cent et a fiancé par Claude Cochet de Genève.
(fol. 146v°)	(fol. 146v°)
(Bory, de Coppet) - Lequelt a infringyr et incour plussieurs poiennes par desobayssance faicte riere Ciilignyez, toutesfoys a prier l'havoyer pour recommandé et le pardoner desdictes offences. Et, ayans aoys le chatellaen de Cillignyez, lequelt a refferuz qui a reparer le biez etc., ordonné que lesdictes poiennes incorues soyent mitigués, pour toutes choses, à cent lyvres monoye.	(Bory de Coppet) - Lequel a enfreint et incour plusieurs peines par désobéissance faite rière Ciiligny toutefois a prié l'avoir pour recommandé et le pardonner desdites offences. Et ayant ouï le châtelain de Céligny lequel a référé qui a réparé le biez etc. ordonné que lesdites peines incorues soient mitigées pour toutes choses à cent livres monnaie.

Figure 4: Post-editing interface used in our experiments to measure the productivity gain brought by automatic normalisation models compared to manually normalising RCs from scratch. The source text is presented on the left side while the normalised hypothesis is presented on the right side. Each editable block contains a segment as it appears in the original manuscript.

Overview of the EvaLatin 2024 Evaluation Campaign

Rachele Sprugnoli[1],Federica Iurescia[2], Marco Passarotti[2]
Università di Parma, viale D'Azeglio, 85, 43125 Parma, Italy[1],
Università Cattolica del Sacro Cuore, largo A. Gemelli 1, 20123 Milan, Italy[2]
rachele.sprugnoli@unipr.it, {federica.iurescia, marco.passarotti}@unicatt.it

Abstract

This paper describes the organization and the results of the third edition of EvaLatin, the campaign for the evaluation of Natural Language Processing tools for Latin. The two shared tasks proposed in EvaLatin 2024, i. e. Dependency Parsing and Emotion Polarity Detection, are aimed to foster research in the field of language technologies for Classical languages. The shared datasets are described and the results obtained by the participants for each task are presented and discussed.

Keywords: Latin, evaluation, dependency parsing, emotion polarity detection

1. Introduction

EvaLatin 2024 is the third edition of the campaign devoted to the evaluation of Natural Language Processing (NLP) tools for the Latin language. As in 2020 (Sprugnoli et al., 2020a) and 2022 (Sprugnoli et al., 2022), EvaLatin is proposed as part of the *Workshop on Language Technologies for Historical and Ancient Languages* (LT4HALA), co-located with LREC COLING 2024.[1] Similar to what happens in other international evaluation campaigns, participants were provided with shared test data that are made freely available for research purposes to encourage further improvement of language technologies for Latin. Shared scripts were also provided. Data, scorer and detailed guidelines are all available in a dedicated GitHub repository.[2]
EvaLatin is an initiative organized by the CIRCSE research centre[3] at the Università Cattolica del Sacro Cuore in Milan, Italy, together with the University of Parma, Italy.

2. Tasks

EvaLatin 2024 is organized around 2 tasks:

- **Dependency Parsing**: the aim of the task is to provide syntactic analysis of Latin texts following the Universal Dependencies (UD) framework (de Marneffe et al., 2021). The output submitted by the participants is a CoNLL-U file with indications of the syntactic head and of the dependency relations in the fields 7 (`HEAD`) and 8 (`DEPREL`) respectively.
- **Emotion Polarity Detection**: the aim of the task is to identify the polarity conveyed by each sentence in the input text, taking into consideration both the vocabulary used by the author and the images that are evoked in the text (Sprugnoli et al., 2023). More specifically, the question to be answered is: which of the following classes best describes how are the emotions conveyed by the poet in the sentence under analysis?
  - `positive`: the only emotions that are conveyed in the text are positive, or positive emotions are clearly prevalent;
  - `negative`: the only emotions that are conveyed in the text are negative, or negative emotions are clearly prevalent;
  - `neutral`: there are no emotions conveyed by the text;
  - `mixed`: lexicon and evoked images produce opposite emotions; it is not possible to find a clearly prevailing emotion polarity.

  Sentences are provided in their original order in the source text.

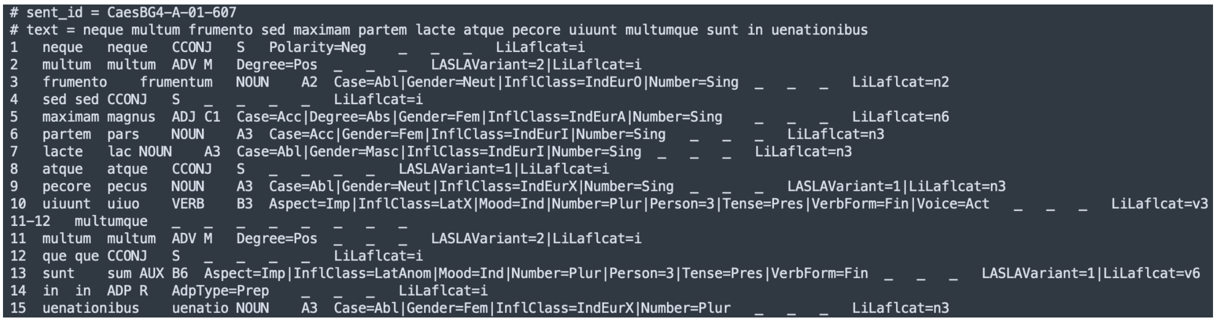
3. Data

No specific training data are released for the Dependency Parsing task but participants are free to make use of any (kind of) resource they consider useful for the task, including the Latin treebanks already available in the UD collection. In this regard, one of the challenges of this task is to understand which treebank (or combination of treebanks) is the most suitable to deal with new test data.

Also for the Emotion Polarity Detection task, no training data are released but an annotation sample and a manually created polarity lexicon are provided. Also in this task, participants are free to

[1] https://lrec-coling-2024.org/
[2] https://github.com/CIRCSE/LT4HALA/tree/master/2024/data_and_doc
[3] https://centridiricerca.unicatt.it/circse/en.html

LT4HALA 2024@LREC-COLING-2024, pages 190–197



```
# sent_id = CaesBG4-A-01-607
# text = neque multum frumento sed maximam partem lacte atque pecore uiuunt multumque sunt in uenationibus
1   neque   neque   CCONJ   S   Polarity=Neg    _   _   _   LiLaflcat=i
2   multum  multum  ADV M   Degree=Pos  _   _   _   LASLAVariant=2|LiLaflcat=i
3   frumento    frumentum   NOUN    A2  Case=Abl|Gender=Neut|InflClass=IndEurO|Number=Sing  _   _   _   LiLaflcat=n2
4   sed sed CCONJ   S   _   _   _   _   LiLaflcat=i
5   maximam magnus  ADJ C1  Case=Acc|Degree=Abs|Gender=Fem|InflClass=IndEurA|Number=Sing    _   _   _   LiLaflcat=n6
6   partem  pars    NOUN    A3  Case=Acc|Gender=Fem|InflClass=IndEurI|Number=Sing   _   _   _   LiLaflcat=n3
7   lacte   lac NOUN    A3  Case=Abl|Gender=Masc|InflClass=IndEurI|Number=Sing  _   _   _   LiLaflcat=n3
8   atque   atque   CCONJ   S   _   _   _   _   LASLAVariant=1|LiLaflcat=i
9   pecore  pecus   NOUN    A3  Case=Abl|Gender=Neut|InflClass=IndEurX|Number=Sing  _   _   _   LASLAVariant=1|LiLaflcat=n3
10  uiuunt  uiuo    VERB    B3  Aspect=Imp|InflClass=LatX|Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act   _   _   _   LiLaflcat=v3
11-12   multumque   _   _   _   _   _   _   _   _
11  multum  multum  ADV M   Degree=Pos  _   _   _   LASLAVariant=2|LiLaflcat=i
12  que que CCONJ   S   _   _   _   _   LiLaflcat=i
13  sunt    sum AUX B6  Aspect=Imp|InflClass=LatAnom|Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin  _   _   _   LASLAVariant=1|LiLaflcat=v6
14  in  in  ADP R   AdpType=Prep    _   _   _   LiLaflcat=i
15  uenationibus    uenatio NOUN    A3  Case=Abl|Gender=Fem|InflClass=IndEurX|Number=Plur   _   _   _   LiLaflcat=n3
```

Figure 1: Example of the test data format.

pursue the approach they prefer, including unsupervised and/or cross-language ones.

Both tasks aim to improve a state of the art that is currently not optimal. With regard to Dependency Parsing, UD treebanks currently show different degrees of harmonization, and Latin is not an exception in this respect (Gamba and Zeman, 2023). With regard to Emotion Polarity Detection, there are no available training data for Latin yet, as this is an unexplored territory for this language. It is important to notice that in both tasks, some texts include punctuation, some do not, as this is the actual state of the art for Latin treebanks and corpora; for example, the LASLA corpus (see Section 3.1 for further details) does not include punctuation (Denooz, 2004). The diversity of the data currently available for both tasks is an issue we are aware of, and that needs to be addressed. This evaluation campaign aims at addressing this issue, and among the desired outcomes there are strategies to deal with it successfully.

3.1. Test Data

Texts provided as test data for the Dependency Parsing task are by 2 Classical authors (Seneca and Tacitus) for a total of more than 13,000 tokens. Each author is taken as specimen of one specific text genre: Seneca for poetry, more specifically for tragedy, with *Hercules Furens* (more than 7,000 tokens), composed in 1st century AD; Tacitus for prose, more specifically historical and ethnographic treatise, with *Germania* (nearly 6,000 tokens), written in 1st century AD. Precise numbers are given in Tables 1 and 2, while an example of the format of test data is given in Figure 1. Data are taken from the LASLA corpus, a linguistic resource manually annotated since 1961 by the Laboratoire d'Analyse Statistique des Langues Anciennes (LASLA) at the University of Liège, Belgium.[4] Original data were converted into the annotation formalism of the UD project and manually annotated for dependency relations. Data are distributed in the CoNLL-U format.[5] Following such format, the annotations are plain text files having the `.conllu` extension and encoded in UTF-8.

AUTHOR	TEXT	#TOKENS
Seneca	Hercules Furens	7,711

Table 1: Test data for poetry.

AUTHOR	TEXT	#TOKENS
Tacitus	Germania	5,669

Table 2: Test data for prose.

Texts provided as test data for the Emotion Polarity Detection task are by 3 authors for a total of 297 sentences (around 100 sentences for each author):

- Seneca, with the final part (lines 1,175-1,344)[6] of the tragedy *Hercules Furens*, composed in 1st century AD;
- Horace, with 16 odes (4 for each book that makes up *Carmina*), composed in 1st century AD;
- Giovanni Pontano, with 12 poems taken from the work *Neniae*, composed in the 15th century.

Test data for the task of Emotion Polarity Detection are distributed in .tsv format: the first column contains a sentence ID and the second the text to be tagged. Tables 3, 4, 5 report the precise number of sentences for each text, while Figure 2 provide an example of the format. Data by Seneca and Horace are taken from the LASLA corpus, while texts by Pontano are taken from the *Poeti d'Italia in*

[4] http://web.philo.ulg.ac.be/lasla/textes-latins-traites/

[5] https://universaldependencies.org/format.html

[6] Line numbers according to the following edition: Fitch, J.G. (2018). *Seneca. Tragedies, Volume I: Hercules. Trojan Women. Phoenician Women. Medea. Phaedra.* Cambridge (MA): Harvard University Press.

lingua latina website.[7] For this reason, Pontano's texts have punctuation while those of Seneca and Horace do not.

AUTHOR	TEXT	#SENT.
Seneca	Hercules Furens (lines 1,175-1,344)	103

Table 3: Test data by Seneca.

AUTHOR	ODE (BOOK_POEM)}	#SENT.
Horace	I_2	7
Horace	I_14	8
Horace	I_28	9
Horace	I_38	2
Horace	II_3	6
Horace	II_11	7
Horace	II_14	3
Horace	II_16	10
Horace	III_2	5
Horace	III_10	4
Horace	III_18	2
Horace	III_24	7
Horace	IV_1	11
Horace	IV_10	1
Horace	IV_12	8
Horace	IV_13	6
TOTAL		**96**

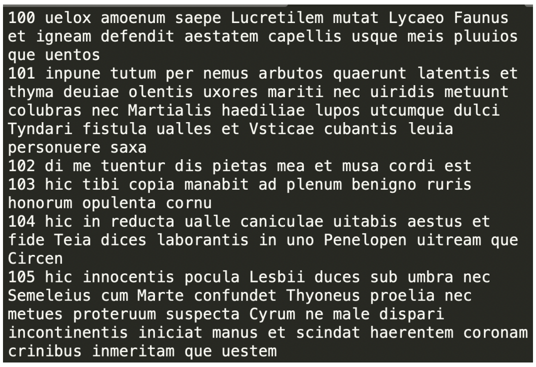
Table 4: Test data by Horace.

AUTHOR	NENIAE	#SENT.
Pontano	I	8
Pontano	II	11
Pontano	III	9
Pontano	IV	14
Pontano	V	6
Pontano	VI	7
Pontano	VII	11
Pontano	VIII	5
Pontano	IX	4
Pontano	X	9
Pontano	XI	8
Pontano	XII	6
TOTAL		**98**

Table 5: Test data by Pontano.

4. Evaluation

Two different scorers are used for the two shared tasks proposed at EvaLatin 2024.



```
100 uelox amoenum saepe Lucretilem mutat Lycaeo Faunus
et igneam defendit aestatem capellis usque meis pluuios
que uentos
101 inpune tutum per nemus arbutos quaerunt latentis et
thyma deuiae olentis uxores mariti nec uiridis metuunt
colubras nec Martialis haediliae lupos utcumque dulci
Tyndari fistula ualles et Vsticae cubantis leuia
personuere saxa
102 di me tuentur dis pietas mea et musa cordi est
103 hic tibi copia manabit ad plenum benigno ruris
honorum opulenta cornu
104 hic in reducta ualle caniculae uitabis aestus et
fide Teia dices laborantis in uno Penelopen uitream que
Circen
105 hic innocentis pocula Lesbii duces sub umbra nec
Semeleius cum Marte confundet Thyoneus proelia nec
metues proteruum suspecta Cyrum ne male dispari
incontinentis iniciat manus et scindat haerentem coronam
crinibus inmeritam que uestem
```

Figure 2: Example of the data format for the Emotion Polarity Detection task.

- The scorer employed for the evaluation of the Dependency Parsing task is the one developed for the *CoNLL18 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies* (Zeman et al., 2018).[8] The evaluation starts by aligning the system-produced tokens to the gold standard one; given that we provide test data already sentence-splitted and annotated with morpho-grammatical information, the alignment for tokens, sentences, words, UPOS, UFeats and lemmas should be perfect (i. e. 100.00). Then, CLAS (Content-Word Labeled Attachment Score)[9] and LAS (Labeled Attachment Score)[10] are evaluated in terms of Precision, Recall, F1 and Aligned Accuracy.[11]
- The scorer for the Emotion Polarity Detection task is a Python script that calculates precision, recall and F1 measure for each class assigned at sentence level but also accuracy, macro-average and weighted average. The scorer is available on the EvaLatin web page[12].

As for the baseline, for the Dependency Parsing

[7] https://www.poetiditalia.it/public/

[8] https://universaldependencies.org/conll18/evaluation.html

[9] CLAS is the labeled F1- score over all relations except those involving function words (aux, case, cc, clf, cop, det, mark) and punctuation (punct). For further details, see (Nivre and Fang, 2017).

[10] LAS is the percentage of tokens assigned both the correct DEPREL and HEAD. For further details, see (Buchholz and Marsi, 2006).

[11] The scorer computes also the Unlabeled Attachment Score (UAS), that is the percentage of tokens assigned the correct HEAD; the Morphology-aware Labeled Attachment Score (MLAS), that is CLAS extended with evaluation of POS tags and morphological features; the Bi-Lexical dependency score (BLEX) that combines content-word relations with lemmatization, but not with POS tags and features. These 3 metrics are not taken into account for this shared task.

[12] https://github.com/CIRCSE/LT4HALA/blob/master/2024/scorer-emotion.py

task we provide the scores obtained on the test data using UDPipe 2 (Straka et al., 2016) with the model trained on the Perseus Universal Dependencies Latin Treebank[13] (Bamman and Crane, 2011), as it is available from the tool's web interface.[14]

For the Emotion Polarity Detection task, we calculate the baseline by applying a lexicon-based approach to the test data. More specifically, a sentence score is computed by summing the polarity values of all lemmas. Polarity values are taken from LatinAffectus v.4, a prior polarity sentiment lexicon for Latin (Sprugnoli et al., 2020b). The label `positive` is assigned to all the sentences with score above 0 and the label `negative` to sentence for which the score is below 0. For scores equal to 0, we attribute `neutral` to sentences where all words have a score of 0 and `mixed` where positive and negative scores are balancing each other out to a total net sum of 0.

5. Results and Discussion

Three teams took part in the Dependency Parsing task and other three teams took part in the Emotion Polarity Detection task. Regarding the latter, one team did not submit the report and therefore it will not be included in this overview.

5.1. Dependency Parsing

Details on the participating teams and their systems for the Dependency Parsing task are given below:

- Behr. This team submitted one run, leveraging historical sentence embeddings generated via SBERT (Reimers and Gurevych, 2019) as a pivotal strategy to confront the challenge of developing a parser capable of achieving accurate performance irrespective of the chronological period of the Latin texts within the test data (Behr, 2024).

- KU Leuven - Brepols CTLO. The team submitted two runs. The first run adopts a span-span prediction methodology, grounded in Machine Reading Comprehension (MRC), and utilizes LaBERTa (Riemenschneider and Frank, 2023), a RoBERTa model pre-trained specifically on Latin corpora. This run yields meaningful outcomes. Conversely, the second, more exploratory run operates at the token-level, employing a span-extraction approach inspired by the Question Answering (QA) task. This model fine-tunes a DeBERTa model (He et al., 2023) pre-trained on Latin datasets, but the results are extremely low (Mercelis, 2024).

- ÚFAL LatinPipe. Also this team submitted two distinct runs employing a system comprising a fine-tuned concatenation of base and large pre-trained Language Models. Both runs utilize a dot-product attention head for parsing and softmax classification heads for morphology, enabling the joint learning of dependency parsing and morphological analysis. Training data are sampled from seven publicly available Latin treebanks, with additional efforts focused on harmonizing annotations to attain a more cohesive annotation style. The difference between the two runs lies in the treatment of punctuation, that is present in some of the treebanks used for the training set, but is absent in the shared test data (Straka et al., 2024).

Table 6 and 7 show the final ranking. The results are provided in terms of F1, including the baseline. The majority of the submitted runs demonstrate clear improvements over the baseline, with the sole exception being the exploratory KU Leuven - Brepols CTLO run 2. Performances remain consistent across diverse text genres (poetry and prose) and evaluation metrics (LAS and CLAS). The best performing run, ÚFAL LatinPipe_1, exhibits a nearly 25% enhancement over the baseline.

The Dependency Parsing task underscores two primary challenges encountered in the development of models for parsing Latin data: firstly, the variability in the annotation styles across available Latin treebanks, posing a challenge to model training; and secondly, the extensive temporal scope and diverse genres present in Latin texts. The teams addressed these challenges relying on Large Language Models (LLMs) to navigate through them effectively. Behr's approach explicitly targets model performance across different epochs, while KU Leuven - Brepols CTLO adopts a span extraction method, drawing inspiration from QA tasks. However, this experimentation reveals limitations in current QA implementations regarding dependency head prediction, indicating the need for further investigation. The ÚFAL LatinPipe team employs LLMs, conducting data harmonization and fine-tuning on various combinations of treebanks, resulting in superior performance.

Presently, leveraging LLMs, fine-tuning on treebank ensembles, and harmonizing inconsistent annotations emerge as the most encouraging strategies for Dependency Parsing in Latin. This shared task demonstrates promising solutions to parsing challenges: harmonization addresses annotation style diversity, while ensemble approaches mitigate portability issues.

[13] https://github.com/UniversalDependencies/UD_Latin-Perseus/

[14] http://lindat.mff.cuni.cz/services/udpipe/

TEAM	F1 POETRY	TEAM	F1 PROSE
ÚFAL LatinPipe_1	74.53	ÚFAL LatinPipe_1	73.19
ÚFAL LatinPipe_2	69.59	ÚFAL LatinPipe_2	68.76
Behr	67.87	Behr	66.53
KU Leuven - Brepols CTLO run 1	57.34	KU Leuven - Brepols CTLO run 1	63.71
BASELINE	48.51	BASELINE	51.81
KU Leuven - Brepols CTLO run 2	5.34	KU Leuven - Brepols CTLO run 2	3.78

Table 6: Dependency Parsing results in terms of CLAS.

TEAM	F1 POETRY	TEAM	F1 PROSE
ÚFAL LatinPipe_1	75.75	ÚFAL LatinPipe_1	77.41
ÚFAL LatinPipe_2	70.68	ÚFAL LatinPipe_2	73.07
Behr	68.33	Behr	69.72
KU Leuven - Brepols CTLO run 1	59.02	KU Leuven - Brepols CTLO run 1	67.32
BASELINE	50.36	BASELINE	56.73
KU Leuven - Brepols CTLO run 2	5.44	KU Leuven - Brepols CTLO run 2	3.70

Table 7: Dependency Parsing results in terms of LAS.

5.2. Emotion Polarity Detection

Details on the participating teams and their systems for the Emotion Polarity Detection task are given below:

- Nostra Domina. This team submitted two runs employing data augmentation algorithms and various Latin LLMs in a neural architecture. Both runs ended up using the same augmentation procedure and LLM, but they differed in their encoder. The first and second runs include a Transformer encoder and BiLSTM encoder, respectively (Bothwell et al., 2024).

- TartuNLP. The team submitted two runs, both based on XLM-RoBERTa, the multilingual version of RoBERTa (Conneau et al., 2020). To deal with the lack of training data, they created two datasets, one by applying LatinAffectus v.4 and the other by using OpenAI's GPT-4. To make the training faster, avoid catastrophic forgetting and capitalize on knowledge transfer, they used parameter efficient fine-tuning methods employing language adapters and multi-stage training. (Dorkin and Sirts, 2024).

Table 8 reports the final ranking, showing the results in terms of F1, including the baseline. Given that Horace and Pontano's test set is made up of various texts, the value reported in the table corresponds to the macro-average F1.

The difficulty of the Emotion Polarity Detection task is evident by looking at the results reported in Table 8. In fact, the baseline is not beaten by every submitted run and it even obtains the best F1 on Pontano's poems. Among the participating systems there is not a single one that performs better than the others on all 3 authors. The TartuNLP_1 run (fine-tuned on a dataset annotated by applying LatinAffectus v.4) is the best performing one on Seneca and Pontano but records the lowest F1 macro-average on Horace for which, on the contrary, the best run is NostraDomina_1 (that uses PhilBERTa-based embeddings (Riemenschneider and Frank, 2023), a Transformer encoder, and a dataset derived from Gaussian clustering). The performances at class level are also different: the NostraDomina team's runs have better results in recognizing positive sentences, while the TartuNLP runs record higher F1 for negative sentences. For all the runs, however, the mixed class is the most difficult to recognize.

In general, there are two important trends that all runs have in common. On the one hand the use of data augmentation methods to make up for the lack of training data, on the other the use of neural models, in particular LLMs.

6. Conclusion

This paper has provided an overview of the NLP tasks addressed in the third edition of the EvaLatin evaluation campaign, namely: Dependency Parsing and Emotion Polarity Detection.

Compared to the tasks of the previous editions of EvaLatin (Lemmatization, PoS tagging, Morphological Feature Identification), the accuracy rates of the tools that participated in the evaluation campaign are lower. This is due both to the higher degree of difficulty of the tasks themselves and to the limited (or nonexistent) availability of training sets to build machine-learning models in a (semi-)supervised manner. To overcome this limitation, the participating systems made extensive use of pre-trained models equipped with knowledge that

TEAM	SENECA	TEAM	HORACE	TEAM	PONTANO
TartuNLP_1	0.26	Baseline	0.40	NostraDomina_1	0.42
Baseline	0.25	TartuNLP_1	0.31	TartuNLP_2	0.32
TartuNLP_2	0.25	TartuNLP_2	0.30	NostraDomina_2	0.31
NostraDomina_2	0.14	NostraDomina_1	0.29	Baseline	0.29
NostraDomina_1	0.12	NostraDomina_2	0.21	TartuNLP_1	0.24

Table 8: Emotion Polarity Detection results in terms of F1.

can be fine-tuned for specific NLP tasks by using the data provided by annotated corpora, which, in an ideal virtuous circle, represent one of the outcomes of the application of NLP tools. In such respect, one of the objectives of EvaLatin was (and still remains) providing a venue for developing and evaluating language models for various NLP tasks to support the building of more and larger annotated corpora for Latin.

The task dedicated to Dependency Parsing has shown that the state of the art is good, although still far from optimal. The problem of model portability across different literary genres, albeit roughly distributed on a binary classification (prose and poetry), remains an open challenge, with a substantial impact on the automatic processing of Latin texts, which exhibit a high degree of stylistic variability.

The task of Emotion Polarity Detection was a risky bet, given the scarcity of external resources that could be used, the absence of training sets, and the lack of previously available annotation guidelines. The low accuracy rates of the participating systems highlight the difficulty of the task, which is also due to the high degree of subjectivity intrinsic to the task itself and to the involvement of many different components (lexical, syntactic, encyclopedic, cultural) in determining the emotion evoked by a text.

Emotion Polarity Detection opens the door for EvaLatin to semantic analysis, which includes tasks such as Semantic Role Labeling and Word Sense Disambiguation. It is our intention to consider these types of NLP tasks for the future editions of the evaluation campaign.

7. Acknowledgements

The authors want to thank Lisa Sophie Albertelli, Lorenzo Augello, Roberta Buffolino, Giulia Calvi, Roberta Leotta and Marinella Testori for the annotation of test data for the Emotion Polarity Detection task and Giovanni Moretti for providing the scorer for the Emotion Polarity Detection task.

8. Bibliographical References

David Bamman and Gregory Crane. 2011. The ancient Greek and Latin dependency treebanks. In *Language Technology for Cultural Heritage*, Theory and Applications of Natural Language Processing, pages 79–98, Berlin/Heidelberg, Germany. Springer. Preprint retrievable at `http://www.cs.cmu.edu/~dbamman/pubs/pdf/latech2011.pdf`.

Rufus Behr. 2024. Behr at EvaLatin 2024: Latin Dependency Parsing Using Historical Sentence Embeddings. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2024)*, Torino, Italy. European Language Resources Association.

Stephen Bothwell, Abigail Swenor, and David Chiang. 2024. Nostra Domina at EvaLatin 2024: Improving Latin Polarity Detection through Data Augmentation. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2024)*, Torino, Italy. European Language Resources Association.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.

Flavio Massimiliano Cecchini. 2021. *Formae reformandae*: for a reorganisation of verb form annotation in Universal Dependencies illustrated by the specific case of Latin. In *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*, pages 1–15, Sofia, Bulgaria. The Association for Computational Linguistics (ACL). Retrievable at `https://aclanthology.org/2021.udw-1.1/`.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 8440–8451, Online. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308. Retrievable at https://direct.mit.edu/coli/article/47/2/255/98516/Universal-Dependencies.

Joseph Denooz. 2004. Opera Latina : une base de données sur internet. *Euphrosyne*, 32:79–88.

Aleksei Dorkin and Kairit Sirts. 2024. TartuNLP at EvaLatin 2024: Emotion Polarity Detection. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2024)*, Torino, Italy. European Language Resources Association.

Federica Gamba and Daniel Zeman. 2023. Universalising Latin Universal Dependencies: a harmonisation of Latin treebanks in UD. pages 7–16, Washington, D.C. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Wouter Mercelis. 2024. KU Leuven / Brepols-CTLO at EvaLatin 2024: Span extraction approaches for Latin dependency parsing. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2024)*, Torino, Italy. European Language Resources Association.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal Dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 86–95, Gothenburg, Sweden. Association for Computational Linguistics.

Marco Passarotti, Francesco Mambrini, Greta Franzini, Flavio Massimiliano Cecchini, Eleonora Litta, Giovanni Moretti, Paolo Ruffolo, and Rachele Sprugnoli. 2020. Interlinking through Lemmas. The Lexical Collection of the LiLa Knowledge Base of Linguistic Resources for Latin. *Studi e Saggi Linguistici*, LVIII(1):177–212. Retrievable at https://www.studiesaggilinguistici.it/ssl/article/view/277.

Caroline Philippart de Foy. 2014. *LASLA – Nouveau manuel de lemmatisation du latin*. LASLA, Liège, Belgium. Retrievable at https://orbi.uliege.be/handle/2268/171931.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Frederick Riemenschneider and Anette Frank. 2023. Exploring large language models for classical philology. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15181–15199, Toronto, Canada. Association for Computational Linguistics.

David A Smith, Jeffrey A Rydberg-Cox, and Gregory R Crane. 2000. The Perseus Project: a digital library for the humanities. *Literary and Linguistic Computing*, 15(1):15–25.

Rachele Sprugnoli, Francesco Mambrini, Marco Passarotti, and Giovanni Moretti. 2023. The Sentiment of Latin Poetry. Annotation and Automatic Analysis of the Odes of Horace. *IJCoL. Italian Journal of Computational Linguistics*, 9(9-1):53–71.

Rachele Sprugnoli, Marco Passarotti, Flavio Massimiliano Cecchini, Margherita Fantoli, and Giovanni Moretti. 2022. Overview of the EvaLatin 2022 evaluation campaign. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 183–188, Marseille, France. European Language Resources Association.

Rachele Sprugnoli, Marco Passarotti, Flavio Massimiliano Cecchini, and Matteo Pellegrini. 2020a. Overview of the EvaLatin 2020 evaluation campaign. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 105–110, Marseille, France. European Language Resources Association (ELRA).

Rachele Sprugnoli, Marco Passarotti, Daniela Corbetta, and Andrea Peverelli. 2020b. Odi et Amo. creating, evaluating and extending sentiment lexicons for Latin. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3078–3086, Marseille, France. European Language Resources Association.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA). Retrievable at https://aclanthology.org/L16-1680.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, BC, Canada. Association for Computational Linguistics.

Milan Straka, Jana Straková, and Federica Gamba. 2024. ÚFAL LatinPipe at EvaLatin 2024: Morphosyntactic Analysis of Latin. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2024)*, Torino, Italy. European Language Resources Association.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Behr at EvaLatin 2024: Latin Dependency Parsing Using Historical Sentence Embeddings

Rufus Behr
Research Computing at Northeastern University
r.behr@northeastern.edu

Abstract

This paper identifies the system used for my submission to EvaLatin's shared dependency parsing task as part of the LT4HALA 2024 workshop. EvaLatin presented new Latin prose and poetry dependency test data from potentially different time periods, and imposed no restriction on training data or model selection for the task. This paper, therefore, sought to build a general Latin dependency parser that would perform accurately regardless of the Latin age to which the test data belongs. To train a general parser, all of the available Universal Dependencies treebanks were used, but in order to address the changes in the Latin language over time, this paper introduces historical sentence embeddings. A model was trained to encode sentences of the same Latin age into vectors of high cosine similarity, which are referred to as historical sentence embeddings. The system introduces these historical sentence embeddings into a biaffine dependency parser with the hopes of enabling training across the Latin treebanks in a more efficacious manner, but their inclusion shows no improvement over the base model.

Keywords: Natural Language Processing (NLP), Dependency Parsing, Latin

1. Introduction

EvaLatin's (Sprugnoli et al., 2024) dependency parsing task, which makes use of the Universal Dependency Parsing framework[1], permitted the use of any models and combination of training data to parse new test data created for this task, consisting of both prose and poetic texts from different time periods. One of the main challenges for this task, therefore, is identifying which combination of treebanks and data to use.

There are two main complications regarding dependency parsing data for Latin: its comparatively low-resource nature and the evolution of the language over time. Nehrdich and Hellwig (2022), citing Passarotti and Ruffolo (2010) and McGillivray and Passarotti (2009), explain that prior works on dependency parsing for Latin have domain transfer issues, where the training on one treebank yields poorer results on others. The authors explain, citing Dinkova-Bruun (2011) and Vincent (2016), that this issue stems from the linguistic evolution of the language over time, which, for instance, can be seen when comparing Classical Latin to Medieval, and this change is reflected in the respective dependency parsing treebanks for those time periods. Consequently, even though Latin is well-studied and has sizable extant text compared to other low-resource languages, the change in the language over time can make it prohibitive to use all the data that is available.

The two most widely-used forms of dependency parsing algorithms are graph-based and transition-based. In the latter, the parser moves across the sentence, adding words to a stack, and, given the top elements of the stack and its prior transitions, predicts if there's a dependency arc (Jurafsky and Martin). Graph-based parsing algorithms, however, encode a given sentence into a fully connected, weighted, and directed graph, where each vertex is a word and each edge a possible relation, and the parser then assigns scores for each edge. Afterwards, they find the maximum spanning tree for this graph, which is deemed the best parse tree (Jurafsky and Martin; Altıntaş and Tantuğ, 2023). A notable downside to transition-based parsing is that it necessarily creates a projective tree (Jurafsky and Martin), whereas graph-based parsing can produce non-projective trees. As Nehrdich and Hellwig (2022) explain, one reason graph-based parsing is preferred for Latin is the freedom of word order, resulting in possibly non-projective dependency trees.

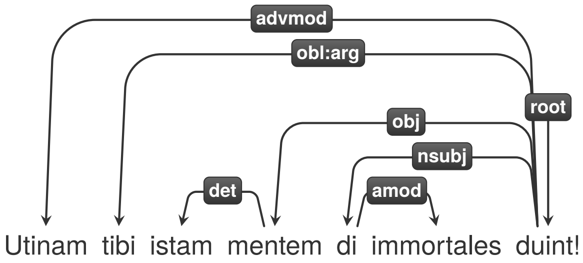


Figure 1: An example of dependency parsed Latin text from Cicero's *in Catilinam*

The predominant neural graph-based dependency parsing architecture comes from Dozat and

[1] www.universaldependencies.org

Manning (2017). This parser takes the words of the sentences and creates 100-dimensional uncased word vectors concatenated with their part-of-speech tag vectors (Dozat and Manning, 2017; Altıntaş and Tantuğ, 2023). These are then processed by three Bidirectional Long Short-Term Memory (BiLSTM) layers, the output of which is passed into four Multilayer perceptrons (MLP). Two of the MLPs are used to identify head and dependent arcs and the other two to identify their labels (Dozat and Manning, 2017; Altıntaş and Tantuğ, 2023). The vectors of the MLPs are passed into two biaffine classifiers, which produce score matrices for the dependency arcs and their label probabilities (Dozat and Manning, 2017; Altıntaş and Tantuğ, 2023).

This architecture by Dozat and Manning (2017) was the base architecture used for the Latin dependency parsing done by Nehrdich and Hellwig (2022), which achieved state-of-the-art results. The authors modified this architecture by employing contextualized Latin word embeddings from Latin BERT (Bamman and Burns, 2020).

This paper uses the dependency parser model architecture and code from Attardi et al. (2021) as its base. That model is a modified version of the semantic dependency parser proposed by Dozat and Manning (2018), which was an extension of the authors' prior work for semantic dependency parsing. The modification by Attardi et al. (2021) was in its loss function, using softmax cross-entropy rather than sigmoid.

The model in this paper builds on top of this architecture by introducing a historical sentence embedding produced by a Sentence-BERT (SBERT) model (Reimers and Gurevych, 2019), trained for this submission. The sentence embedding is concatenated with the output of the BiLSTM before being passed into the four MLPs. This embedding is introduced with the hope that the model might yield better results when trained on the Latin treebanks that span different periods in the history of Latin.

2. Model Architecture and Resources

Universal Dependencies has five Latin treebanks available: Index Thomisticus Treebank (ITTB) (Passarotti, 2019), Late Latin Charter Treebank (LLCT) (Cecchini et al., 2020b), Perseus (Bamman and Crane, 2011), UDante treebank (Cecchini et al., 2020a), and PROIEL (Haug and Jøhndal, 2008).

Gamba and Zeman (2023) experimented with a new workflow that involves harmonising all the Universal Dependency Latin treebanks before training with UDPipe and Stanza, but they found that the parsing accuracy only improved slightly after applying the harmonisation process.

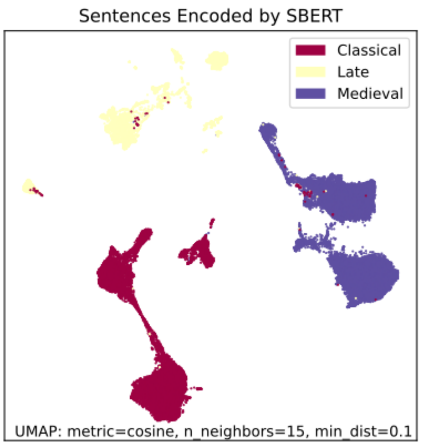


Figure 2: The data encoded using the SBERT model and then visualised with UMAP

This paper also makes use of all the available Universal Dependency Latin treebanks — in particular, the Universal Dependency version 2.13 release of ITTB (Passarotti, 2019), LLCT (Cecchini et al., 2020b), Perseus (Bamman and Crane, 2011), UDante (Cecchini et al., 2020a), and PROIEL (Haug and Jøhndal, 2008), but rather than modification of the treebanks prior to training, it introduces a historical sentence embedding. This idea is inspired by the work done by Altıntaş and Tantuğ (2023), where they show improved dependency parsing performance through concatenating features, including sentence representation, to the tokens before the MLP layer. The authors did not use SBERT themselves, but they did list it as a possible sentence representation.

Filename	Kept Sentences	Sentences Skipped
la_udante-ud-train.conllu	926	0
la_udante-ud-dev.conllu	375	1
la_udante-ud-test.conllu	419	0
la_ittb-ud-test.conllu	1879	222
la_ittb-ud-dev.conllu	1936	165
la_ittb-ud-train.conllu	21107	1668
la_llct-ud-dev.conllu	752	98
la_llct-ud-train.conllu	6189	1100
la_llct-ud-test.conllu	715	169
la_proiel-ud-train.conllu	15515	681
la_proiel-ud-test.conllu	1201	59
la_proiel-ud-dev.conllu	1171	62
la_perseus-ud-train.conllu	1324	10
la_perseus-ud-test.conllu	935	4
Total	54444	4239

Table 1: Treebank data for the experiments, loaded and displayed sequentially

In preparation for both the dependency parsing training and the SBERT training, the five treebanks were merged with exact duplicates removed. To identify these duplicates, sentences were compared against previously processed sentences, and if a new sentence's text was found previously, it was skipped in the merging process. Table 1 shows the data loaded in order during the merging process.

Age	Number of Sentences	Percentage
Classical	19192	49%
Late	8610	16%
Medieval	26642	35%

Table 2: Sentence Distribution by Age

In addition to removing duplicates, the sentences are sorted by their Latin age, the resulting distribution of which can be seen in Table 2.

2.1. Historical Sentence Embedding

As the intended purpose of including a historical sentence embedding is guiding the parser dependent on the text's corresponding Latin age to allow training on all treebanks, encoded sentences of the same age should have a higher cosine similarity, whereas sentences of other ages should be dissimilar.

As stated, this paper uses SBERT (Reimers and Gurevych, 2019), a fine-tuned version of BERT — in this case Latin BERT (Bamman and Burns, 2020) — designed for encoding sentences, to create these historical sentence embeddings. To prepare the training data, 50,000 random unique sentence pairs were selected from the five treebanks' 54,444 unique sentences, and each pair was assigned a similarity label: 1.0 if the authors were the same, .8 if they were from the same Latin age, and 0.0 if they were from different Latin ages.

The SBERT model, trained on that data, was able to embed sentences into 256-dimensional vectors, where sentences of the same Latin age are similar. You can see the historical sentence embeddings of the data from the five treebanks, mapped to two dimensions using UMAP (McInnes et al., 2020), in Fig. 2.

2.2. Model Architecture

The parser architecture, as described at the end of Section 1, is a modification of the dependency parser from Attardi et al. (2021) with the notable incorporation of the SBERT model, trained as described in Section 2.1.

Given a sentence, the model creates the historical sentence embedding and then creates the word embeddings and Latin BERT (Bamman and Burns, 2020) embeddings, which are concatenated together. These embeddings are then passed through the BiLSTMs, whose outputs are then concatenated with the historical sentence embedding. At this point, the values are run through the MLPs and biaffine classifiers as in the base model.

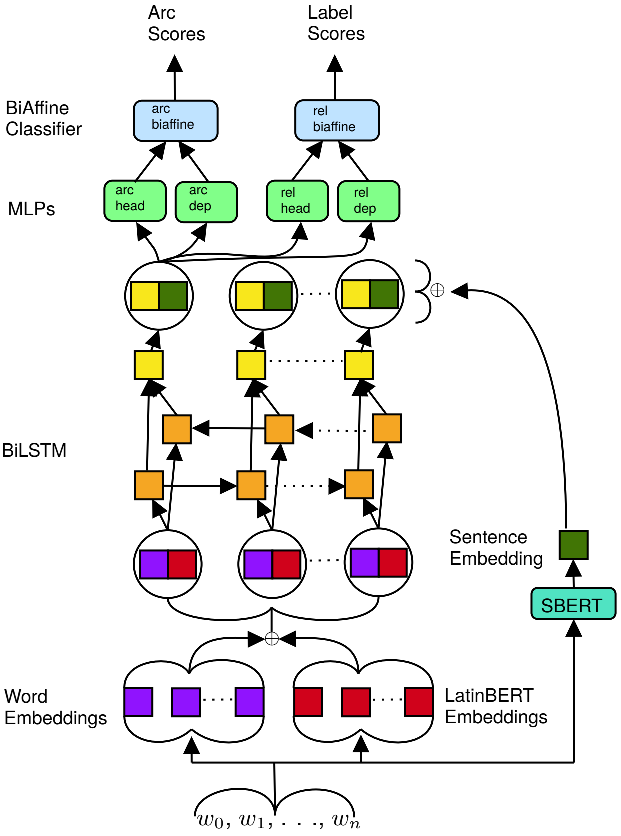


Figure 3: The Model Architecture

The model's architecture diagram can be seen in Fig. 3 with the omission of layer repetition (i.e., there are three BiLSTM layers, but the figure shows only one).

The data for training the model was selected through stratified sampling from the merged treebanks with respect to the Latin age. The training data uses 67% of the total data, and the remaining 33% were split evenly between test and development sets.

3. Experiments

Three models were selected for the experiments: the proposed model with SBERT, trained on the 5 treebanks; the base diaparser model (Attardi et al., 2021), trained on the 5 treebanks; and the pretrained Latin diaparser model, which was trained on ITTB (Passarotti, 2019) and LLCT (Cecchini et al., 2020b).

The only change to the hyperparameters from the original diaparser implementation was to change Adam's epsilon value to 1e-6 from the original 1e-12.

These three models were evaluated on the test data created for EvaLatin using the provided script[2]

[2] https://github.com/CIRCSE/LT4HALA/blob/master/2024/conll18_ud_eval.py

		Poetry				Prose			
Model	Metric	Precision	Recall	F1	AligndAcc	Precision	Recall	F1	AligndAcc
Diaparser with SBERT	CLAS	67.31	68.45	67.87	68.45	66.31	66.74	66.53	66.74
	LAS	68.33	68.33	68.33	68.33	69.72	69.72	69.72	69.72
	UCM				35.50				14.38
	LCM				15.14				3.68
Diaparser without SBERT	CLAS	67.33	68.59	67.95	68.59	65.83	66.60	66.21	66.60
	LAS	68.28	68.28	68.28	68.28	68.28	68.28	68.28	68.28
	UCM				33.87				14.38
	LCM				13.69				3.68
Pretrained Diaparser	CLAS	24.35	24.11	24.23	24.11	33.26	33.29	33.27	33.29
	LAS	26.45	26.45	26.45	26.45	39.39	39.39	39.39	39.39
	UCM				2.34				2.34
	LCM				0.00				0.0

Table 3: The results of the different models evaluated on the EvaLatin gold conllu files

to find the CLAS and LAS, and then the UCM and LCM were found for each model using diaparser's built-in evaluate function[3].

All of the code for the experimentation and data preparation is available on GitHub[4].

4. Results and Analysis

The results presented in Table 3 are a combination of the official ones for the EvaLatin submission, which used the historical embeddings, and subsequent evaluation runs done using the same script with the performance metrics as described in Section 3.

The results show no significant improvement with the inclusion of the historical sentence embedding proposed. Both models that were trained on the totality of the text provided did outperform the pretrained model, which is likely reflective of the lack of Classical Latin text in the model's training dataset compared to its proportion in the EvaLatin test set.

5. Conclusions

This paper experimented with the application of a historical Latin sentence embedding to help guide a Latin dependency parser, inspired by Altıntaş and Tantuğ (2023). Although the inclusion of this sentence embedding did not improve the overall performance of the parser, future research might focus on the inclusion of other features to guide Latin graph-based dependency parsing to enable better training across the treebanks.

6. Acknowledgements

This work was completed in part using the Discovery cluster, supported by Northeastern University's Research Computing team.

7. Bibliographical References

Mücahit Altıntaş and A. Cüneyd Tantuğ. 2023. Improving the performance of graph based dependency parsing by guiding bi-affine layer with augmented global and local features. *Intelligent Systems with Applications*, 18:200190.

Giuseppe Attardi, Daniele Sartiano, and Maria Simi. 2021. Biaffine dependency and semantic graph parsing for EnhancedUniversal dependencies. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 184–188, Online. Association for Computational Linguistics.

David Bamman and Patrick J. Burns. 2020. Latin BERT: A contextual language model for classical philology. *CoRR*, abs/2009.10053.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Federica Gamba and Daniel Zeman. 2023. Latin morphology through the centuries: Ensuring consistency for better language processing. In *Proceedings of the Ancient Language Processing Workshop*, pages 59–67, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

[3] https://github.com/Unipisa/diaparser/?tab=readme-ov-file#evaluation

[4] https://github.com/SufurElite/LatinDependencyParser

Dan Jurafsky and James H. Martin. *Chapter 18: Dependency Parsing*, 3rd edition, pages 5–6,15–16.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction.

Sebastian Nehrdich and Oliver Hellwig. 2022. Accurate dependency parsing and tagging of Latin. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 20–25, Marseille, France. European Language Resources Association.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Rachele Sprugnoli, Federica Iurescia, and Marco Passarotti. 2024. Overview of the EvaLatin 2024 evaluation campaign. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages â€“ LT4HALA 2024*, Torino, Italy. European Language Resources Association.

8. Language Resource References

Bamman, David and Crane, Gregory. 2011. *The Ancient Greek and Latin Dependency Treebanks*. Springer Berlin Heidelberg, Theory and Applications of Natural Language Processing.

Flavio Cecchini, Rachele Sprugnoli, Giovanni Moretti, and Marco Passarotti. 2020a. Udante: First steps towards the universal dependencies treebank of dante's latin works.

Cecchini, Flavio Massimiliano and Korkiakangas, Timo and Passarotti, Marco. 2020b. *A New Latin Treebank for Universal Dependencies: Charters between Ancient Latin and Romance Languages*. European Language Resources Association.

Dag Trygve Truslew Haug and Marius L. Jøhndal. 2008. Creating a parallel treebank of the old indo-european bibletranslations.

Passarotti, Marco. 2019. *The Project of the Index Thomisticus Treebank*.

KU Leuven / Brepols-CTLO at EvaLatin 2024: Span extraction approaches for Latin dependency parsing

Wouter Mercelis
KU Leuven / Brepols-CTLO
KU Leuven: Blijde Inkomststraat 21, B-3000 Leuven, Belgium
Brepols-CTLO: Begijnhof 39, B-2300 Turnhout, Belgium
wouter.mercelis@kuleuven.be

Abstract

This report describes the KU Leuven / Brepols-CTLO submission to EvaLatin 2024. We present the results of two runs, both of which try to implement a span extraction approach. The first run implements span-span prediction, rooted in Machine Reading Comprehension, while making use of LaBERTa, a RoBERTa model pretrained on Latin texts. The first run produces meaningful results. The second, more experimental run operates on the token-level with a span-extraction approach based on the Question Answering task. This model finetuned a DeBERTa model, pretrained on Latin texts. The finetuning was set up in the form of a Multitask Model, with classification heads for each token's part-of-speech tag and dependency relation label, while a question answering head handled the dependency head predictions. Due to the shared loss function, this paper tried to capture the link between part-of-speech tag, dependency relation and dependency heads, that follows the human intuition. The second run did not perform well.

Keywords: Latin, NLP, dependency parsing, span extraction, question answering

1. Introduction

This short report describes the two runs of the KU Leuven / Brepols-CTLO team for the EvaLatin 2024 Evaluation Campaign (Sprugnoli, Iurescia and Passarotti, 2024), specifically for the Latin dependency parsing task. For each of the dependency parsing runs, this report will discuss the methodology (including the pre-trained language model), the actual results and a short discussion of the results.

2. MRC-based span-span prediction

2.1 Methodology

One of the first aims of our run was to look for an alternative to Dozat and Manning's (2017) Biaffine parser. Gan et al. (2022) propose a two-step method, called MRC-based span-span prediction, which firstly tries to predict subtrees in a dependency tree of a sentence, and secondly predicts the links between these proposed subtrees. The authors claim state-of-the-art performance on various benchmarks. In addition to this, the method also works with non-projective dependency trees, which is important for languages with a relatively free word order such as Latin.

Gan et al's (2022) method requires a pretrained language model as a starting point. We opted for the RoBERTa-like LaBERTa (Riemenschneider and Frank, 2023) for the following reasons. Firstly, we encountered some technical difficulties using our own DeBERTa-based model, as the tokenizer approach of Gan et al. (2022) was not compatible with our DeBERTa-based model. Due to time constraints, we decided to switch to a model with broader support. Furthermore, we chose LaBERTa because the original paper performed best with a similar XLM-RoBERTa (Conneau *et al.*, 2020) model. Therefore, we decided to use a model which has an equivalent architecture and training process.

For the training data, we took advantage of the work of Gamba and Zeman (2023), in which harmonization measures were introduced to reduce the disparity between the five Latin Universal Dependencies (UD) (de Marneffe *et al.*, 2021) treebanks. We opted to train on the Perseus (UD v2.13)(Bamman and Crane, 2011) and the ITTB (UD v2.13) (Passarotti, 2019) treebanks, as the Perseus treebank aligns the most with the test data. The ITTB treebank is mainly included because of its large size.

Concerning training parameters, we used the default parameters out-of-the-box, with a reduced batch size of 4 to prevent CUDA out-of-memory errors.

2.2 Results

	Poetry			
	Precision	Recall	F1	AligndAcc
CLAS	57.26	57.42	57.34	57.42
LAS	59.02	59.02	59.02	59.02
	Prose			
	Precision	Recall	F1	AligndAcc
CLAS	63.93	63.49	63.71	63.49
LAS	67.32	67.32	67.32	67.32

Table 1: KU Leuven/Brepols-CTLO run 1 results

LT4HALA 2024@LREC-COLING-2024, pages 203–206

	Poetry			
	Precision	Recall	F1	AligndAcc
CLAS	74.34	74.72	74.53	74.72
LAS	75.75	75.75	75.75	75.75
	Prose			
	Precision	Recall	F1	AligndAcc
CLAS	73.58	72.80	73.19	72.80
LAS	77.41	77.41	77.41	77.41

Table 2: ÚFAL LatinPipe_1 results

In Table 1, the results of our first run are summarized, while in Table 2, the results of the best-performing team are shown in comparison.

2.3 Discussion

To start with, the Chu-Liu-Edmonds algorithm failed once to generate a proper graph, resulting in a dependency tree with two roots, which is not well-formed. This was solved by considering the second root as a conjunction of the first one.

Apart from this slight mishap, the results were quite disappointing. For a large part, this can be explained by a misinterpretation of the guidelines from our part. As the guidelines contained information about all the main relations, and referred to the UD website for more information about the subrelations, we wrongly interpreted this as supplementary information, meaning that the subrelations would not be taken into account during evaluation. This had a considerable impact on our accuracy numbers. For example, almost half of the wrongly predicted dependency relation labels contained a subrelation in the gold data (913 out of 1871 wrong predictions).

Another problematic notion are coordinating constructions. For the 805 “conj” instances in the prose gold data, in 305 cases the wrong head was predicted. Similarly, for the 605 “cc” instances, 175 receive a wrong head relation. The same method reveals that 96 of the 299 roots do not receive the correct head relation. This is can possibly be attributed to differences in annotation between test and training data. Furthermore, with regards to ellipsis in clauses, the UD framework prefers assigning the root label to non-verbs in verb-final languages such as Latin. Our model has trouble taking this into account, preferring to use the final verb as a root instead.

3. Multitask Question Answering

3.1 Methodology

Our second run was much more experimental. During work on word alignment, we used a span extraction approach that is also used in Question Answering. As an experiment, we tried to apply this naively to dependency parsing as well. In fact, the first run can be seen as a more elaborate approach to this problem, in a way that is more suited to the task as well.

For this second run, we made use of a Multitask Model, in which a pretrained language model is finetuned using different classification heads, with a shared loss function, as shown in Figure 1. For a theoretical survey, see Crawshaw (2020). Due to this shared loss function, the model is not only very efficient, it also quantifies the learning of inter-task dependencies and generalizes well, following our intuition that the relation labels, the relations themselves and the part-of-speech tags all influence each other.

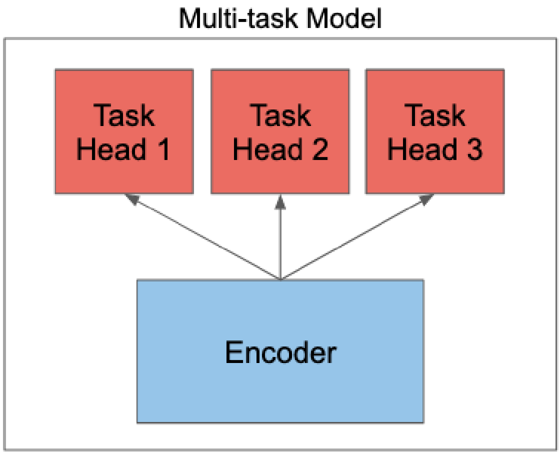


Figure 1: Architecture of a multi-task model

In this task we used our own DeBERTa-model (He et al., 2023). Starting with DeBERTa v3, the pretraining approach is very similar to the ELECTRA approach (Clark *et al.*, 2020), with better results. This Latin DeBERTa model is the successor of the ELECTRA model that we used in the 2022 Evalatin Competition (Mercelis and Keersmaekers, 2022). It is also trained on Brepols’ Library of Latin Texts[1], in addition to various online corpora such as the CAMENA project[2] and web data such as the Latin Wikisource[3] and Wikipedia[4]. As copyright rests on the Library of Latin Texts, this model is not publicly available.

[1] https://www.brepols.net/series/LLT-O
[2] https://mateo.uni-mannheim.de/camenahtdocs/camena_e.html
[3] https://la.wikisource.org/wiki/Pagina_prima
[4] https://la.wikipedia.org/wiki/Vicipaedia:Pagina_prima

For the finetuning data, we experimented with only ITTB (UD v2.13) (Bamman and Crane, 2011) as our training data in the first place, planning to add more data if the experiments were fruitful. Unfortunately, these addition have not yet taken place due to time constraints.

As the multitask model performed well during experiments with jointly predicting morphological tags, we tried extending it to dependency parsing as well. Crucially, this task is fundamentally different from morphological tagging in the sense that on the one hand, tokens cannot be predicted in a vacuum: they are inherently part of a sentence. On the other hand, due to the nature of our span extraction task, we have to input the tokens one by one. By contrast, in token classification tasks, the input is an entire sentence of tokens that are predicted in one go.

This has severe complications for the training process of our model. Table 3 shows in a simplified way how our model processes the data. For clarity, the same sentence is used in the example. Note that during the experiment, this data is shuffled at random, so the same sentence will be spread throughout the data for each of the finetuning tasks.

Tokens	Training task
[CLS] unde et dicit … [SEP] [PAD] ...	token classification (POS)
[CLS] unde et dicit … [SEP] [PAD] ...	token classification(deprel)
[CLS] [SEP] unde [SEP] et dicit … [SEP] [PAD] ...	question answering, first token
[CLS] unde [SEP] et [SEP] dicit … [SEP] [PAD] ...	question answering, second token
[CLS] unde et [SEP] dicit [SEP] … [SEP] [PAD] ...	question answering, third token

Table 3: Overview of the training data structure

As the data are shuffled at random, the part-of-speech tagging, the dependency relations and the dependency heads are not learned at the same stage in the training process.

Adding to this, we encountered more technical difficulties, as said above, resulting in a batch size of 1, which is also not ideal. We did not have enough time to have an in-depth look into these issues. Also, due to these time constraints, we could not try this approach with the LaBERTa model as well.

3.2 Results

	Poetry			
	Precision	Recall	F1	AligndAcc
CLAS	5.36	5.33	5.34	5.33
LAS	5.44	5.44	5.44	5.44
	Prose			
	Precision	Recall	F1	AligndAcc
CLAS	3.79	3.76	3.78	3.76
LAS	3.70	3.70	3.70	3.70

Table 4: KU Leuven/Brepols-CTLO run 2

Table 4 shows the results of our second run. See table 2 for a comparison with the results of the best-performing team.

3.3 Discussion

As seen above, the results are not meaningful at all. Unexpectedly, the model performs worse on prose than on poetry. However, the obtained results are so low that this does not tell anything about the performance of the model. In fact, we only included this run so we could discuss the architecture *in se*. We could see that the implementation of the Chu-Liu-Edmonds algorithm had difficulties providing a meaningful graph, resulting in many sentences with multiple predicted roots. We used the same algorithm as in the previous model to reduce them to well-formed sentences. This however resulted in many wrongly predicted heads. However, the dependency relation labels did not suffer from this approach at all. For the prose data, 4402 tokens out of 5840 received the right dependency relation label, outperforming our first run, which labeled only 3969 tokens correctly. This leads us to believe that the multi-task approach is not the problem, but rather the current question-answering implementation that predicts the dependency heads.

Thus, we believe that with a proper technical implementation, there is something to say for this approach. However, the focus needs to shift from the token level to the sentence level.

4. Conclusion

In conclusion, our first run performed reasonably well, unfortunately hampered by the subrelation issue. This shows that there are performant alternatives to Dozat and Manning's Biaffine parser. Our second run did not perform well, but can serve as a building block for further research, as this multi-task model shows promise especially in the prediction of dependency labels.

5. Acknowledgments

Our work has been funded by grant no. HBC.2021.0210 of Flanders Innovation and Entrepreneurship.

6. Bibliographical References

Clark, K. *et al.* (2020) 'ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators', *arXiv:2003.10555 [cs]* [Preprint]. Available at: http://arxiv.org/abs/2003.10555 (Accessed: 4 October 2021).Conneau, A. et al. (2020) 'Unsupervised Cross-lingual Representation Learning at Scale', in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online: Association for Computational Linguistics, pp. 8440–8451. Available at: https://doi.org/10.18653/v1/2020.acl-main.747.

Crawshaw, M. (2020) 'Multi-Task Learning with Deep Neural Networks: A Survey'. arXiv. Available at: http://arxiv.org/abs/2009.09796 (Accessed: 10 March 2024).

Dozat, T. and Manning, C.D. (2017) 'Deep Biaffine Attention for Neural Dependency Parsing', *arXiv:1611.01734 [cs]* [Preprint]. Available at: http://arxiv.org/abs/1611.01734 (Accessed: 19 March 2021).

Gamba, F. and Zeman, D. (2023) 'Universalising Latin Universal Dependencies: a harmonisation of Latin treebanks in UD', in L. Grobol and F. Tyers (eds) *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*. *UDW-SyntaxFest 2023*, Washington, D.C.: Association for Computational Linguistics, pp. 7–16. Available at: https://aclanthology.org/2023.udw-1.2 (Accessed: 10 March 2024).

Gan, L. *et al.* (2022) 'Dependency Parsing as MRC-based Span-Span Prediction', in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, pp. 2427–2437. Available at: https://doi.org/10.18653/v1/2022.acl-long.173.

He, P., Gao, J. and Chen, W. (2023) 'DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing'. arXiv. Available at: https://doi.org/10.48550/arXiv.2111.09543.

Mercelis, W. and Keersmaekers, A. (2022) 'An ELECTRA Model for Latin Token Tagging Tasks', in R. Sprugnoli and M. Passarotti (eds) *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*. *LT4HALA 2022*, Marseille, France: European Language Resources Association, pp. 189–192. Available at: https://aclanthology.org/2022.lt4hala-1.30 (Accessed: 10 March 2024).

Riemenschneider, F. and Frank, A. (2023) 'Exploring Large Language Models for Classical Philology', in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada: Association for Computational Linguistics, pp. 15181–15199. Available at: https://doi.org/10.18653/v1/2023.acl-long.846.

Sprugnoli, R., Iurescia, F. and Passarotti, M. (2024) 'Overview of the EvaLatin 2024 Evaluation Campaign', in R. Sprugnoli and M. Passarotti (eds) *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages @LT4HALA 2024*. Torino, Italy: European Language Resources Association.

7. Language Resource References

Bamman, D. and Crane, G. (2011) 'The Ancient Greek and Latin Dependency Treebanks', in C. Sporleder, A. van den Bosch, and K. Zervanou (eds) *Language Technology for Cultural Heritage*. Heidelberg: Springer, pp. 79–98. Available at: https://doi.org/10.1007/978-3-642-20227-8_5.

de Marneffe, M.-C. *et al.* (2021) 'Universal Dependencies', *Computational Linguistics*, 47(2), pp. 255–308. Available at: https://doi.org/10.1162/coli_a_00402.

Passarotti, M. (2019) 'The Project of the Index Thomisticus Treebank', in M. Berti (ed.) *Digital Classical Philology. Ancient Greek and Latin in the Digital Revolution*. Berlin - Boston: De Gruyter Saur, pp. 299–320. Available at: https://doi.org/10.1515/9783110599572-017.

ÚFAL LatinPipe at EvaLatin 2024: Morphosyntactic Analysis of Latin

Milan Straka, Jana Straková, Federica Gamba
Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics
Charles University, Czech Republic
{straka,strakova,gamba}@ufal.mff.cuni.cz

Abstract

We present LatinPipe, the winning submission to the EvaLatin 2024 Dependency Parsing shared task. Our system consists of a fine-tuned concatenation of base and large pre-trained LMs, with a dot-product attention head for parsing and softmax classification heads for morphology to jointly learn both dependency parsing and morphological analysis. It is trained by sampling from seven publicly available Latin corpora, utilizing additional harmonization of annotations to achieve a more unified annotation style. Before fine-tuning, we train the system for a few initial epochs with frozen weights. We also add additional local relative contextualization by stacking the BiLSTM layers on top of the Transformer(s). Finally, we ensemble output probability distributions from seven randomly instantiated networks for the final submission. The code is available at https://github.com/ufal/evalatin2024-latinpipe.

Keywords: dependency parsing, part of speech tagging, EvaLatin, Latin, LatinPipe

1. Introduction

In this paper, we describe our entry to the EvaLatin 2024 Dependency Parsing shared task (Sprugnoli et al., 2024). Our system is called LatinPipe to resemble its predecessors, UDPipe (Straka and Straková, 2017) and UDPipe 2 (Straka, 2018). We submitted two variants, called *ÚFAL LatinPipe 1* and *ÚFAL LatinPipe 2*, placing 1st and 2nd in the shared task evaluation, respectively.

Our system is an evolution of UDPipe 2 (Straka, 2018). LatinPipe is a graph-based dependency parser which uses a deep neural network for scoring the graph edges. Unlike UDPipe 2, the neural network architecture of LatinPipe is a fine-tuned pre-trained language model, with a dot-product attention head for dependency parsing and softmax classification heads for morphological analysis to learn both these tasks jointly.

We provide an extensive evaluation of the approaches used in LatinPipe: a comparison of monolingual and multilingual pre-trained language models and their concatenations; initial pretraining on the frozen Transformer weights; adding two BiLSTM layers on top of the Transformers; and using the gold UPOS from the shared task data on the network input. A considerable focus is directed at multi-treebank training, as well as the harmonization of annotation styles among the seven publicly available Latin treebanks.

2. Related Work

The EvaLatin 2024 Dependency Parsing shared task (Sprugnoli et al., 2024) builds upon the two previous editions of EvaLatin, which focused respectively on lemmatization and POS tagging (Sprugnoli et al., 2020) and lemmatization, POS tagging, and features identification (Sprugnoli et al., 2022). UDPipe 2 won the EvaLatin 2020 shared task (Straka and Straková, 2020); previously, it participated in the 2018 CoNLL Shared Tasks on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2018), which encompassed also Latin, and placed among the winning systems (Straka, 2018).

Latin Dependency Parsing In recent years, Nehrdich and Hellwig (2022) developed a graph-based dependency parser specifically for Latin. Their approach modifies the architecture of the biaffine parser proposed by Dozat and Manning (2017) by incorporating a character-based convolutional neural network (CharCNN), and exploits Latin BERT embeddings (Bamman and Burns, 2020).

Fantoli and de Lhoneux (2022) trained a POS tagging and parsing model using the deep biaffine parser (Dozat and Manning, 2017) implementation of MaChAmp (van der Goot et al., 2021) and exploiting treebank embeddings in the encoder.

Karamolegkou and Stymne (2021) explored Latin parsing in a low-resource scenario and found ancient Greek to be most effective as transfer language, likely due to its syntactic similarity with Latin.

3. Data

Latin Treebanks We train LatinPipe on the training portions of the following seven publicly available Latin corpora:

- ITTB of UD 2.13 (Passarotti, 2019);
- LLCT of UD 2.13 (Cecchini et al., 2020a);

Corpus	Training tokens
ITTB	391K
LLCT	194K
PROIEL	178K
UDante	31K
Perseus	18K
Sab	11K
Arch	1K
UD 2.13	812K
UD 2.13+Sab+Arch	824K

Table 1: Training data sizes in tokens.

- PROIEL in either of these two versions: UD 2.13 (Haug and Jøhndal, 2008), and a UD-style harmonized version (Gamba and Zeman, 2023a,b);[1]
- UDante of UD 2.13 (Cecchini et al., 2020b);
- Perseus of UD 2.13 (Bamman and Crane, 2011);
- UD-style annotated text of *De Latinae Linguae Reparatione* by Marcus Antonius Sabellicus (Gamba and Cecchini, 2024);
- *Archimedes Latinus* UD-style treebank (Fantoli and de Lhoneux, 2022), based on the Latin translation of the Greek mathematical work *The Spirals* of Archimedes;[2]

where UD 2.13 stands for the Universal Dependencies project (Nivre et al., 2020), version 2.13 (Zeman et al., 2023). We denote the former five corpora distributed by UD 2.13 as *UD 2.13* and all seven corpora including additionally *Arch* and *Sab* as *UD 2.13+Arch+Sab* in our experiments. The treebank training data sizes are presented in Table 1.

For the shared task, we train in multi-treebank setting, in which the examples from the abovementioned corpora are sampled into training batches proportionally to the square root of the number of their sentences, similarly to van der Goot et al. (2021).

Harmonization of Annotation Styles We noticed that the PROIEL treebank stands out most in terms of annotation style from the rest of the other treebanks, so much so that the differences in annotation style result in varying performance. We therefore experimented with the following three settings:

- training with a harmonized version of PROIEL by Gamba and Zeman (2023a,b), submitted as *ÚFAL LatinPipe 1*;
- training without PROIEL altogether, submitted as *ÚFAL LatinPipe 2*;
- training with the original PROIEL annotation by Haug and Jøhndal (2008), not submitted due to the two-runs-per-team limit.

The harmonized version of PROIEL resulted from the harmonization carried out by Gamba and Zeman (2023a,b), who observed persisting differences in the annotation scheme of the five Latin treebanks, annotated by different teams and in different stages of the development of UD guidelines. Divergences were observed at all annotation levels, from word segmentation to lemmatization, POS tags, morphology, and syntactic relations. The implemented harmonization process led to substantial improvements in parsing performances, confirming the need for a truly standardized annotation style. Notably, among the five treebanks, in the case of PROIEL a lower degree of accordance with the UD guidelines was observed. For instance, in compound numerals like *viginti quattuor* 'twenty-four' the second number is attached to the first one through a `fixed` relation; in the harmonized version, such dependencies are reannotated as `flat`. Moreover, PROIEL makes use of the `dep` relation, intended for cases where a more precise deprel cannot be assigned. Through POS tags and morphology, in the harmonized version `dep` is replaced with a more appropriate one.

4. Methods

LatinPipe is a graph-based dependency parser. First, a deep learning neural network is used to score the graph edge values, and then a global optimization Chu-Liu/Edmonds' algorithm (Chu and Liu, 1965; Edmonds, 1967) for finding the minimum spanning tree problem is run on the graph.

For scoring the graph edge values, LatinPipe pursues a deep learning approach and consists of a fine-tuned pre-trained LM (or a concatenation of them) with a dot-product parsing attention head. In addition, morphology softmax classification heads are also used, so LatinPipe jointly learns both dependency parsing and morphological analysis.

The general overview of the architecture is given in Figure 1 and the details are outlined in the following paragraphs.

Pre-trained LMs Our baselines are either fine-tuned LaBerta or PhilBerta, the Latin monolingual RoBERTa base language models by Riemenschneider and Frank (2023); or the fine-tuned XLM-RoBERTa large (Conneau et al. (2020); 355M parameters), which was pretrained on 390M

[1] Available for download at `https://github.com/fjambe/Latin-variability/tree/main/morpho_harmonization/morpho-harmonized-treebanks`.

[2] Available at `https://github.com/mfantoli/ArchimedesLatinus`.

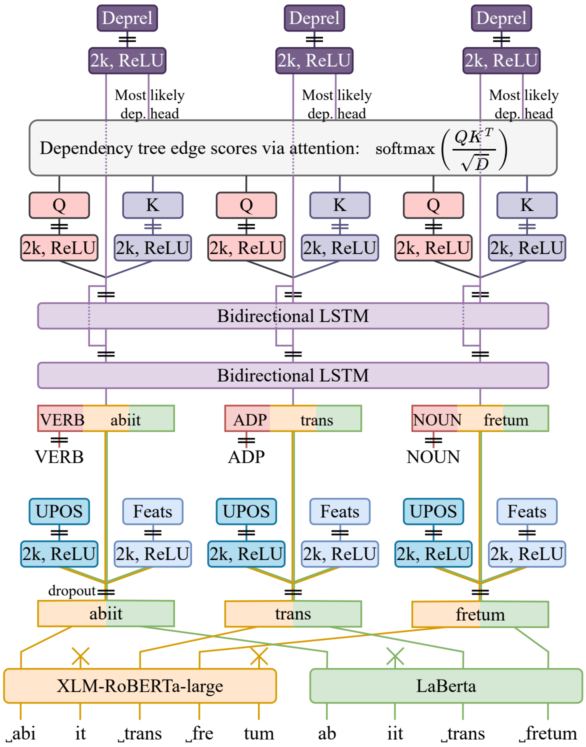


Figure 1: LatinPipe architecture overview.

Latin tokens among other languages. Apart from using the single fine-tuned PLMs, we also experimented with a concatenation of the contextualized embeddings yielded by multiple fine-tuned encoders: *LaBerta+PhilBerta* and *XLM-R large+LaBerta+PhilBerta*.

Frozen Pretraining Before fine-tuning the PLMs' weights, we can optionally freeze the pre-trained Transformer weights, and optimize solely the remaining weights of the architecture for a few initial epochs, namely the heads and the stacked BiLSTM layers. The objective of frozen pretraining is to facilitate the commencement of the fine-tuning optimization from a favorable starting point.

Adding LSTMs We incorporate two bidirectional LSTM layers (Hochreiter and Schmidhuber, 1997; Gers et al., 2000) on top of the Transformer(s) to enhance the modeling of relative short-distance relationships between the tokens and to contextualize the embedded UPOS tags.

Gold UPOS on Input We leverage the gold morphological analysis provided in the shared task data as an additional input to the neural network. The trainable word embeddings of UPOS are concatenated with the contextualized embeddings yielded from the fine-tuned PLM(s), and together, the concatenation of embeddings is processed by the LSTM layers.

Ensembling For the final submission, we ensemble output probability distributions from seven randomly instantiated networks by averaging the probabilities in the corresponding dimensions.

Handling punctuation The shared task test data do not contain punctuation. This causes concern in settings when training without PROIEL, which is the only representative of a treebank without punctuation. Training solely on data containing punctuation is expected to lead to inferior performance on test data without it. Therefore in this particular setting, we artificially add punctuation to the test data by appending periods at sentence ends, and after the model prediction, we remove the dummy punctuation again.[3]

Architecture Details In the LatinPipe architecture (Figure 1), every classification layer and computation of queries and keys is preceded by a hidden layer of size 2 048 with ReLU activation. The dimensionality of the queries and keys is 512, and the LSTM dimensionality is 256. When predicting dependency relations, we also concatenate the LSTM-generated representation of the most likely dependency head according to the predicted scores (which is not necessarily the one chosen by the Chu-Liu/Edmonds' algorithm).

Training Details The model is trained with the Adam optimizer (Kingma and Ba, 2015) for 30 epochs, each comprising 1 000 batches with a batch size of 32. The learning rate is first linearly increased from 0 to 2e-5 in the first two epochs and then decays to 0 according to the cosine schedule. Optionally, we perform 10-epoch pretraining with frozen Transformer weights utilizing a constant learning rate of 1e-3. On a single A100 GPU with 40GB, the training takes 9 hours. The exact training configuration, including the exact command used to train the models, is available in the released source code.

5. Results

We present the evaluation on the UD 2.13 test data in Tables 2 and on the EvaLatin 2024 test data in Table 3. All the results are averages of three runs.

Table 2.A evaluates the baseline fine-tuned PLMs on the UD 2.13 test sets. Increasing PLM size from base to large clearly improves the results across the board and on average, even if the large model is not a monolingual but a multilingual one.

[3] Obviously, the other option would be to remove the punctuation from the training data and retrain the models, an expensive and unavailable option due to the restricted time span of the shared task testing period.

Experiment	Avg	ITTB	LLCT	PROIEL	UDante	Perseus
A) PLMs Evaluation						
LaBerta	83.20	90.91	94.54	86.75	66.71	77.08
PhilBerta	82.87	91.09	94.19	86.13	66.42	76.51
LaBerta+PhilBerta	83.99	91.31	94.74	87.29	68.18	**78.42**
XLM-R large	84.19	91.60	95.33	87.18	71.17	75.67
XLM-R large+LaBerta+PhilBerta	**84.67**	**91.78**	**95.35**	**87.57**	**71.95**	76.70
B) Incremental Architecture Improvements w.r.t. the Previous Row						
+ Frozen training for 10 epochs	86.09	92.29	95.34	88.64	74.20	79.98
+ Two bidirectional LSTM layers	86.33	92.81	94.70	89.05	74.78	80.32
+ Gold UPOS on parser input	**86.97**	**93.18**	**95.64**	**89.78**	**74.99**	**81.28**
C) Multi-treebank Training w.r.t. the Previous Row						
Single-treebank training	86.97	**93.18**	**95.64**	**89.78**	74.99	81.28
UD 2.13 training	88.05	92.25	95.60	88.74	79.84	**83.84**
UD 2.13+Sab+Arch training	**88.09**	92.18	95.44	88.43	**80.56**	83.81
D) Ensembles of the Models in the Previous Section						
Single-treebank training, 7 models	87.31	**93.38**	95.78	**90.23**	75.51	81.66
UD 2.13 training, 7 models	88.51	92.65	**95.89**	89.10	80.91	84.02
UD 2.13+Sab+Arch training, 7 models	**88.63**	92.45	95.78	89.23	**81.47**	**84.22**
E) Previous work						
UDPipe 2 (Straka, 2018), UD 2.12		*89.35*	*94.39*	*79.55*	*68.65*	*71.91*
MaChAmp (van der Goot et al., 2021), UD 2.8		*92.45*	*95.41*	*86.97*	*74.01*	*74.67*
Nehrdich and Hellwig (2022), UD 2.8-2.9		*92.99*	—	*86.34*	—	*80.16*

Table 2: UD 2.13 test sets LAS evaluation. Avg denotes the LAS macro average over the UD 2.13 corpora. Section E shows previous work on older UD versions.

Experiment	Avg	Poetry	Prose
A) Single-treebank Training			
ITTB	59.96	57.84	62.08
LLCT	47.93	45.12	50.74
PROIEL original	68.87	68.47	69.26
PROIEL harmonized	**73.88**	**72.37**	**75.40**
UDante	60.23	59.11	61.36
Perseus	59.22	58.43	60.02
B) Multi-treebank with PROIEL Versions			
UD 2.13, original	72.31	72.10	72.52
UD 2.13, none	66.16	64.03	68.29
UD 2.13, harmonized	75.22	**74.65**	75.78
UD 2.13+Sab+Arch, original	72.75	72.35	73.14
UD 2.13+Sab+Arch, none	66.64	64.50	68.79
UD 2.13+Sab+Arch, harmo.	**75.48**	74.52	**76.43**
C) Multi-treebank w/ and wo/ Gold UPOS			
w/ gold UPOS	**75.48**	**74.52**	**76.43**
wo/ gold UPOS	74.19	73.28	75.09
D) Ensembles of 7 Models			
UD 2.13+Sab+Arch, original	73.76	73.57	73.95
UD 2.13+Sab+Arch, none	68.16	65.71	70.60
UD 2.13+Sab+Arch, harmo.	**76.58**	**75.75**	**77.41**
E) Adding Punctuation Before Prediction			
UD 2.13+Sab+Arch, none	71.87	70.68	73.07

Table 3: EvaLatin 2024 test set LAS evaluation. Avg denotes the LAS macro average over Poetry and Prose.

The only exception is Perseus, on which we suspect the XLM-R large to overtrain due to the small size of the corpus (see Table 1). Finally, a concatenation of models yields further gains over their single components in all cases.

Table 2.B shows a notable macro average gain of +1.42 percent points when pretraining with frozen weights for initial 10 epochs before fine-tuning. Also the addition of the two bidirectional LSTM layers helps marginally on average by +0.24. Unsurprisingly, the addition of gold UPOS on input brings +0.64 percent points in the UD 2.13 macro average, as well as it improves performance in all single UD 2.13 treebanks. On the EvaLatin test set, the addition of the gold UPOS straightforwardly improved the results by +1.2 on Poetry and +1.3 on Prose, as measured on the non-ensembled model (Table 3.C).

Table 2.C compares multi-treebank training vs. single-treebank training. In accord with previous literature (Nehrdich and Hellwig, 2022), we observed the greatest benefits from the multi-treebank training for the smaller datasets (UDante and Perseus), indecisive results for the middle-sized datasets (LLCT and PROIEL), and a decrease for the largest dataset (ITTB). However, in macro average, we gained +0.51 percent point by multi-treebank training. While the addition of the two new small datasets, the Sab and Arch, is indecisive on the

Experiment	Avg	ITTB	LLCT	PROIEL	UDante	Perseus
A) Best Single-model Results						
Single-treebank training	**97.33**	**99.37**	**99.77**	**98.32**	**93.61**	95.55
UD 2.13 training	97.23	99.25	**99.77**	98.10	93.18	**95.85**
B) Best 7-Model Ensemble Results						
Single-treebank training, 7 models	**97.43**	**99.39**	99.78	**98.47**	**93.61**	95.89
UD 2.13 training, 7 models	97.42	99.33	**99.79**	98.31	93.58	**96.09**
C) Previous work						
UDPipe 2 (Straka, 2018), UD 2.12		*99.03*	*99.75*	*97.02*	*92.95*	*91.18*
MaChAmp (van der Goot et al., 2021), UD 2.8		*98.62*	*99.68*	*97.84*	*91.44*	*90.46*
Nehrdich and Hellwig (2022), UD 2.8-2.9		*97.3*	—	*94.2*	—	*90.8*
Bamman and Burns (2020), UD 2.6		*98.8*	—	*98.2*	—	*94.3*

Table 4: UD 2.13 test sets UPOS evaluation, with Avg denoting the UPOS macro average.

Experiment	Avg	ITTB	LLCT	PROIEL	UDante	Perseus
A) Best Single-model Results						
Single-treebank training	92.45	**98.57**	97.33	**94.68**	83.06	88.61
UD 2.13 training	**93.68**	98.26	**97.36**	94.05	**88.27**	**90.49**
B) Best 7-Model Ensemble Results						
Single-treebank training, 7 models	92.68	**98.62**	97.42	**95.04**	83.37	88.94
UD 2.13 training, 7 models	**94.19**	98.45	**97.52**	94.56	**89.16**	**91.24**
C) Previous work						
UDPipe 2 (Straka, 2018), UD 2.12		*97.12*	*97.16*	*91.43*	*84.38*	*84.65*
MaChAmp (van der Goot et al., 2021), UD 2.8		*96.95*	*96.79*	*92.56*	*69.72*	*84.32*

Table 5: UD 2.13 test sets UFeats evaluation, with Avg denoting the UFeats macro average.

UD 2.13 macro average in Table 2.C, which is in alignment with their modest size (Table 1), on EvaLatin 2024 (Table 3.B), we observed a marginal improvement when incorporating Sab and Arch, which might probably be attributed to similarity of the EvaLatin test data to these treebanks.

Table 3 shows the evaluation on the EvaLatin test data, both Poetry and Prose, and their LAS macro average; with focus on the effect of data harmonization. In all paired experiments, the harmonized PROIEL version clearly improved results over the version with the original PROIEL dataset from UD 2.13, when evaluated on the EvaLatin 2024 test data. However, using at least the original PROIEL dataset in the multi-treebank training is still better than excluding the PROIEL treebank altogether.

As evidenced by both Table 2.D and Table 3.D, an ensemble is always stronger than its individual components. Ensembling adds on average +0.45 percent points on the UD 2.13 LAS macro average over three experimental settings (compare sections C and D in Table 2). In the shared task, ensembling adds +1.26 percent points (compare sections B and D in Table 3). Our best entry, submitted as *ÚFAL LatinPipe 1*, corresponds to the row *UD 2.13+Sab+Arch, harmo.* in Table 3.D.

Finally, when training without PROIEL in a multi-treebank setting, we have to mitigate the punctuation mismatch between the training and the shared task test data, as described in Section 4. Row *UD 2.13+Sab+Arch* in Table 3.E shows our second submission to the shared task, *ÚFAL LatinPipe2*, in which we corrected for missing punctuation in the shared task test data.

UPOS and UFeats Tagging Since our model performs full morphosyntactic analysis, we present also the accuracy of UPOS tagging and UFeats tagging in Tables 4 and 5, respectively. LatinPipe surpasses the previous systems and sets new state-of-the-art results for all treebanks.

6. Conclusion

We described LatinPipe, the winning entry to the EvaLatin 2024 Dependency Parsing shared task, and we provided the evaluation and rationale behind our system design choices. The source code for LatinPipe is available at `https://github.com/ufal/evalatin2024-latinpipe`. Our future work will entail drawing insights from the methodologies presented in this context for the development of UDPipe 3.

7. Acknowledgements

This work has been supported by the Grant Agency of the Czech Republic under the EXPRO program as project "LUSyD" (project No. GX20-16819X), and by the Grant Agency of Charles University as project GAUK No. 104924 "Adapting Uniform Meaning Representation (UMR) for the Italic/Romance languages". The work described herein uses resources hosted by the LINDAT/CLARIAH-CZ Research Infrastructure (projects LM2018101 and LM2023062, supported by the Ministry of Education, Youth and Sports of the Czech Republic).

8. Bibliographical References

David Bamman and Patrick J. Burns. 2020. Latin BERT: A contextual language model for classical philology. *CoRR*, abs/2009.10053.

David Bamman and Gregory Crane. 2011. The Ancient Greek and Latin Dependency Treebanks. In *Language Technology for Cultural Heritage*, pages 79–98, Berlin, Heidelberg. Springer Berlin Heidelberg.

Flavio Massimiliano Cecchini, Timo Korkiakangas, and Marco Passarotti. 2020a. A New Latin treebank for Universal Dependencies: Charters between Ancient Latin and Romance Languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 933–942, Marseille, France. European Language Resources Association.

Flavio Massimiliano Cecchini, Rachele Sprugnoli, Giovanni Moretti, and Marco Passarotti. 2020b. UDante: First Steps Towards the Universal Dependencies Treebank of Dante's Latin Works. In *Proceedings of the Seventh Italian Conference on Computational Linguistics*, pages 1–7. Italian Association for Computational Linguistics (AILC).

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14(10):1396–1400.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking Embedding Coupling in Pre-trained Language Models. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *5th International Conference on Learning Representations*, pages 1–8.

Jack Edmonds. 1967. Optimum Branchings. *Journal of Research of the National Bureau of Standards, B*, 71(4):233–240.

Margherita Fantoli and Miryam de Lhoneux. 2022. Linguistic Annotation of Neo-Latin Mathematical Texts: A Pilot-Study to Improve the Automatic Parsing of the Archimedes Latinus". In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 129–134, Marseille, France. European Language Resources Association.

Federica Gamba and Daniel Zeman. 2023a. Latin Morphology through the Centuries: Ensuring Consistency for Better Language Processing. In *Proceedings of the Ancient Language Processing Workshop*, pages 59–67, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Federica Gamba and Daniel Zeman. 2023b. Universalising Latin Universal Dependencies: a harmonisation of Latin treebanks in UD. In *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, pages 7–16, Washington, D.C. Association for Computational Linguistics.

Felix Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural computation*, 12(10):2451–2471.

Dag Trygve Truslew Haug and Marius Jøhndal. 2008. Creating a Parallel Treebank of the Old Indo-European Bible Translations. In *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, pages 27–34.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Antonia Karamolegkou and Sara Stymne. 2021. Investigation of Transfer Languages for Parsing Latin: Italic Branch vs. Hellenic Branch. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 315–320, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Sebastian Nehrdich and Oliver Hellwig. 2022. Accurate Dependency Parsing and Tagging of Latin. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 20–25, Marseille, France. European Language Resources Association.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Marco Passarotti. 2019. The Project of the Index Thomisticus Treebank. *Digital Classical Philology*, 10:299–320.

Frederick Riemenschneider and Anette Frank. 2023. Exploring Large Language Models for Classical Philology. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15181–15199, Toronto, Canada. Association for Computational Linguistics.

Rachele Sprugnoli, Federica Iurescia, and Marco Passarotti. 2024. Overview of the EvaLatin 2024 Evaluation Campaign. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages LT4HALA 2024*, Torino, Italy. European Language Resources Association.

Rachele Sprugnoli, Marco Passarotti, Flavio Massimiliano Cecchini, Margherita Fantoli, and Giovanni Moretti. 2022. Overview of the EvaLatin 2022 Evaluation Campaign. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 183–188, Marseille, France. European Language Resources Association.

Rachele Sprugnoli, Marco Passarotti, Flavio Massimiliano Cecchini, and Matteo Pellegrini. 2020. Overview of the EvaLatin 2020 evaluation campaign. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 105–110, Marseille, France. European Language Resources Association (ELRA).

Milan Straka. 2018. UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2020. UDPipe at EvaLatin 2020: Contextualized Embeddings and Treebank Embeddings. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 124–129, Marseille, France. European Language Resources Association (ELRA).

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive Choice, Ample Tasks (MaChAmp): A Toolkit for Multi-task Learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael

Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

9. Language Resource References

Gamba, Federica and Cecchini, Flavio Massimiliano. 2024. *De Latinae Linguae Reparatione treebank*. PID http://hdl.handle.net/11234/1-5438. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Zeman, Daniel and Nivre, Joakim and others. 2023. *Universal Dependencies 2.13*. PID http://hdl.handle.net/11234/1-5287. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Nostra Domina at EvaLatin 2024: Improving Latin Polarity Detection through Data Augmentation

Stephen Bothwell, Abigail Swenor, David Chiang
University of Notre Dame
Notre Dame, Indiana, USA
{sbothwel, aswenor, dchiang}@nd.edu

Abstract

This paper describes submissions from the team Nostra Domina to the EvaLatin 2024 shared task of emotion polarity detection. Given the low-resource environment of Latin and the complexity of sentiment in rhetorical genres like poetry, we augmented the available data through automatic polarity annotation. We present two methods for doing so on the basis of the k-means algorithm, and we employ a variety of Latin large language models (LLMs) in a neural architecture to better capture the underlying contextual sentiment representations. Our best approach achieved the second highest macro-averaged Macro-F_1 score on the shared task's test set.

Keywords: emotion polarity detection, sentiment analysis, data augmentation, Latin, LLMs

1. Introduction

Emotion polarity detection is a variant on the common NLP task of sentiment analysis. Usual applications of this task tend to be on reviews—for example, about movies (Maas et al., 2011; Socher et al., 2013) or products (Blitzer et al., 2007)—where providing an opinion is the author's goal. Few works have extended this task to less direct modalities of sentiment, like poetry, and even fewer to ancient languages, like Latin (Chen and Skiena, 2014; Marley, 2018; Sprugnoli et al., 2020, 2023). Thus, the EvaLatin 2024 evaluation campaign's take on this task (Sprugnoli et al., 2024) tackles both an uncommon genre and a low-resource environment.

Motivated by the lack of sentiment resources, this work presents two methods for the automatic annotation of data: *polarity coordinate* clustering, a novel specialization on k-means clustering, and Gaussian clustering. Furthermore, our work examines a variety of different Latin LLMs in a straightforward neural architecture through a hyperparameter search to determine their efficacy on the emotion polarity detection task. To our knowledge, we are the first outside of the original authors to explicitly apply some of these language models for Latin.[1]

After we introduce the small set of pre-existing data for this task, we describe our clustering-based annotation methods (Section 2.1) and their results (Section 2.1.3). Then, we describe our neural architecture (Section 3) and the procedure used for model training and selection (Section 4.1). Finally, we go over our results for this task and investigate why our models performed as they did, with one achieving the second best macro-averaged Macro-F_1 score on EvaLatin's test set (Section 5).

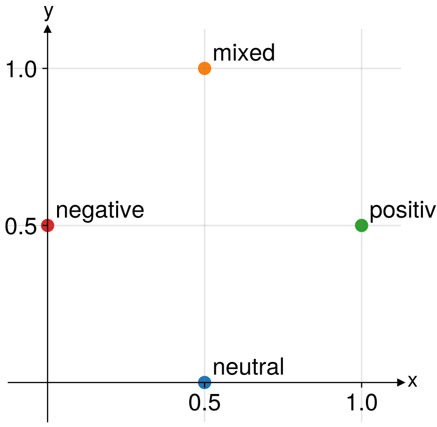


Figure 1: The polarity coordinate plane. Points are all colored differently to represent their classes and are labeled accordingly. The x-axis and y-axis represent polarity and intensity, respectively.

2. Data

Very little data exists for sentiment analysis in Latin. Until recently, only static representations of sentiment were available in sentiment lexica. To our knowledge, the first Latin sentiment lexicon was one automatically transferred to Latin based on English lexica and a large knowledge graph (Chen and Skiena, 2014). This was followed by two others. One was manually curated by a single author based on Stoic values in a study on Cicero (Marley, 2018). The other, called LatinAffectus, was created by multiple Latin experts and organized according to inter-annotator agreement (Sprugnoli et al., 2020); its most recent version, LatinAffectus-v4, was released for use in this shared task.

Following this, Sprugnoli et al. released the first dataset for Latin sentiment analysis. This dataset, having the same classes as our shared task, covers

[1] We make our data and code available at: https://github.com/Mythologos/EvaLatin2024-NostraDomina.

LT4HALA 2024@LREC-COLING-2024, pages 215–222
25 May, 2024.

Dataset	**Class**			
	Positive	Negative	Neutral	Mixed
Odes	20	12	3	9
PC	10427	4114	57786	4178
Gaussian	33473	14333	16861	11838
Horace	20	55	8	15
Pontano	48	18	10	22
Seneca	7	81	2	13
Total	75	154	20	50

Table 1: Resource class distributions. The top, middle, and bottom sections (broken up by pairs of lines) concern pre-existing resources, new resources, and EvaLatin test subsets (or the total set), respectively. “PC” is Polarity Coordinate.

a selection of Horace’s *Odes*—a staple of classical poetry. It contains 44 labeled sentences and has the class distribution given in Table 1. Although this dataset lays groundwork for future studies in Latin sentiment analysis, it is not large enough to train a traditional neural classifier. This is especially the case for a genre which indirectly conveys opinions: poetry frequently employs allusion (*e.g.*, to contemporary circumstances) and rhetorical devices (*e.g.*, metaphor, sarcasm) to make its points.

Given this lack of available training data, we investigate automatic annotation to approximate sentiment for Latin. Because of the variety of time periods, genres, and additional annotations covered by the Universal Dependency (UD) (de Marneffe et al., 2021) treebanks for Latin, we select each of the Perseus (Smith et al., 2000; Bamman and Crane, 2011), PROIEL (Haug et al., 2009), ITTB (Passarotti, 2019), LLCT (Cecchini et al., 2020a), and UDante (Cecchini et al., 2020b) treebanks for this purpose. We also incorporate data from EvaLatin 2022 (Sprugnoli et al., 2022) and the Archimedes Latinus treebank (Fantoli and de Lhoneux, 2022).

2.1. Automatic Annotation

In this section, we detail our data augmentation methods. Both methods relate to the k-means clustering algorithm, where central points—*centroids*—are selected, and the distances between a data point and these centroids relate them in some way.

2.1.1. Polarity Coordinate (PC) Clustering

The task of emotion polarity detection, for the available Latin sentiment data, categorizes each sentence into one of four classes: *positive*, *negative*, *neutral*, and *mixed* (Sprugnoli et al., 2023). This set of classes stems from the circumplex model of affect (Russell and Mehrabian, 1977; Russell, 1980) in which emotions are plotted on a two-dimensional plane with the axes of pleasure-displeasure and arousal-sleep. Sentiment analysis works have often applied this theory with varying terminology (Tian et al., 2018). In our case, we use *polarity* to refer to the “direction” of sentiment (*i.e.*, pleasing or displeasing) and *intensity* to refer to the *magnitude* of the sentiment (*i.e.*, aroused or inert).

These definitions of polarity and intensity can be used to differentiate the four classes for our task. For a given sentence, if its polarity is definitively pleasing, then it is positive; if its polarity is definitively displeasing, then it is negative; if its polarity has both positive and negative elements and has high intensity, then it is mixed; and if it fits into none of these categories (*i.e.*, there is no moderate intensity in either direction), then it is neutral. We employ this mapping to classify sentences via the k-means algorithm. To do so, we must determine the representation for our classes as centroids and our sentences as data points.

Following the idea of the circumplex model, we establish polarity and intensity on a coordinate plane. However, we map the space of these values between 0 and 1, meaning that the point (0.5, 0.5) represents a point of average polarity and intensity. This point is equidistant from each of the four designated class centroids, which we present in Figure 1. Although the positive and negative classes have no innate relation to intensity, we assume that some intensity must exist for the polarity to be noticeable. Given these centroids, we then define a polarity coordinate P for a sequence **x** as:

$$\mathrm{P}_{\mathbf{x}} = (\mathrm{polarity}(\mathbf{x}), \mathrm{intensity}(\mathbf{x})) \tag{1}$$

$$\mathrm{polarity}(\mathbf{x}) = \left(\frac{1}{2|\mathbf{x}|} \sum_{i=1}^{|\mathbf{x}|} \mathrm{score}(\mathrm{x}_i) \right) + \frac{1}{2} \tag{2}$$

$$\mathrm{intensity}(\mathbf{x}) = \frac{1}{|\mathbf{x}|} \sum_{i=1}^{|\mathbf{x}|} |\mathrm{score}(\mathrm{x}_i)| \tag{3}$$

and `score` outputs values between -1 and 1.

To classify sentences, we used LatinAffectus-v4 as the crux of our scoring function. Each $\mathrm{x}_i \in \mathbf{x}$ was searched in the lexicon. To search the lexicon, we used lemmata from the treebank sentences if they were available and the `LatinBackoffLemmatizer` from the Classical Language Toolkit (CLTK) as a backoff option (Johnson et al., 2021).[2] To prevent the impact of sentiment words from being diminished due to the fact that the majority of words were not found in the lexicon, we only used words in LatinAffectus-v4 to score each sentence. This meant that the `polarity` and `intensity` functions

[2] While not necessary for most treebank data, the Archimedes Latinus treebank (Fantoli and de Lhoneux, 2022) does not provide lemmata.

would receive a filtered $\mathbf{x}'$ rather than $\mathbf{x}$. Sentences with no lexical entries were deemed neutral.

Although this method was inspired by the task structure, we suspected that its outputs would be noisy, as it employed static sentiment representations. To account for the noise, we attempted to use the distances between a sentence and each centroid to our advantage. Suppose that we have a collection of distances **d**. We normalized these distances **d**; call this set of normalized distances $\mathbf{d}'$. Then, we calculated a value α for each sentence by subtracting $\min(\mathbf{d}')$ from 1. This α serves as a confidence value for the given label. If the distance for a sentence and its label is low, then the sentence may be a stronger representative for that class and can aid more in the learning process.

With this in mind, we augmented the traditional cross-entropy loss function with a set of these α values, forming what we call the *gold distance weighted cross-entropy* (GDW-CE) loss. Given predictions $\mathbf{Y}'$ and ground truth values $\mathbf{Y}$ (where $|\mathbf{Y}'| = |\mathbf{Y}| = \mathrm{N}$), confidence values $\boldsymbol{\alpha}$, and the cross-entropy function H, the equation for this loss is:

$$\mathbf{GDW\text{-}CE}(\mathbf{Y}', \mathbf{Y}, \boldsymbol{\alpha}) = \sum_{\mathrm{i}=0}^{\mathrm{N}} (\alpha_\mathrm{i} * \mathrm{H}(\mathrm{Y}'_\mathrm{i}, \mathrm{Y}_\mathrm{i})) \qquad (4)$$

2.1.2. Gaussian Clustering

Unlike k-means clustering, Gaussian clustering does not serve as an explicit classifier; instead, it outputs the probabilities for which a given data point is within each cluster. Naturally, however, we can take the cluster with the highest probability to be the label for any given data point. Once again, then, what remains is to establish how the class and sentence representations are derived.

To derive class representations, we trained a Gaussian Mixture Model (GMM) drawn from four distributions (*i.e.*, classes) on the *Odes* dataset (Sprugnoli et al., 2023). We fitted a GMM with the `scikit-learn` library (Virtanen et al., 2020) via the expectation-maximization algorithm. To gather representations for each sentence, we computed sentence-level embeddings from the SPhilBERTa model (Riemenschneider and Frank, 2023b), a pretrained language model for English, Latin, and Ancient Greek based on the Sentence-BERT architecture (Reimers and Gurevych, 2019). We appended the polarity coordinate features described in Section 2.1.1 to these embeddings.

We performed a hyperparameter grid search to select the best GMM. Due to space considerations, we defer the details of this search to our repository. Because of the available data's size, trials were both trained and evaluated on the *Odes* for their Macro-F_1 score; the best GMM scored 0.37.

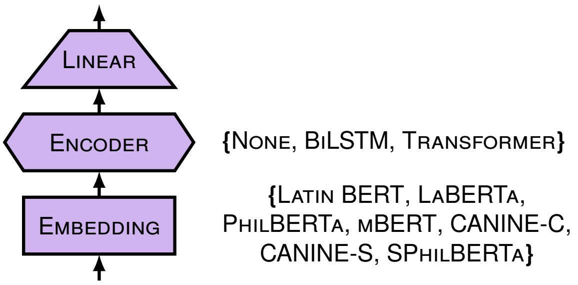

Linear
Encoder
Embedding
{None, BiLSTM, Transformer}
{Latin BERT, LaBERTa, PhilBERTa, mBERT, CANINE-C, CANINE-S, SPhilBERTa}

Figure 2: Architectural options fixed across hyperparameter search trials. Shapes reflect the relative dimensionality of data throughout the network.

2.1.3. Annotation Results

The outcomes of both annotation methods are provided in the middle of Table 1. The PC and Gaussian datasets have dissimilar distributions, preferring the neutral and positive classes, respectively.

3. Modeling

We apply a basic neural architecture for the emotion polarity detection task. As Figure 2 depicts, there are three main parts to this architecture: the embedding, encoder, and linear layers. For the embedding and encoder layers, we have alternatives for each which we examine in our experiments.

For our embeddings, we use all known publicly-available encoder-based LMs containing Latin. Latin BERT (Bamman and Burns, 2020), LaBERTa and PhilBERTa (Riemenschneider and Frank, 2023a), and SPhilBERTa (Riemenschneider and Frank, 2023b) are all either monolingual models (in the case of Latin BERT and LaBERTa) or classical trilingual models (PhilBERTa and SPhilBERTa). We also used the multilingual mBERT (Devlin et al., 2019) and the character-based CANINE-C and CANINE-S (Clark et al., 2022), trained with character-based and subword-based losses, respectively. We froze embeddings during training to maintain their contextual representations.

For our encoders, we employ an identity transformation, a bidirectional LSTM (BiLSTM) (Graves and Schmidhuber, 2005), and a Transformer (Vaswani et al., 2017). For the BiLSTM, we concatenate the final hidden states for both directional LSTMs to provide the final state for classification. For the identity layer and the Transformer, we select the `[CLS]` token's representation.

4. Experiments

4.1. Experimental Design

We divided our annotated data into three splits for training, validation, and testing. The sets contained 80% (61,204 examples), 10% (7,651 examples),

Embedding	Encoder		
	Identity	LSTM	Transformer
Latin BERT	0.12†	0.21*	0.12†
LaBERTa	0.03*	0.17*	0.21†
PhilBERTa	0.06*	0.15*	0.13†
mBERT	0.07†	0.09†	0.08†
CANINE-C	0.14*	0.20†	0.03*
CANINE-S	0.08†	0.17*	0.18†
SPhilBERTa	0.23*	–	–

Table 2: *Odes* Macro-F_1 scores for models trained with data annotated with PC clustering. Since two loss functions were applied per embedding-encoder pair, we show only each pair's maximum score. Values with a $*$ use cross entropy loss, whereas values with a $\dagger$ use GDW-CE loss.

and 10% (7,650 examples) of the overall data, respectively. We used the validation data during training to permit early stopping, setting Macro-F_1 as our criterion of interest with a patience of 10. Otherwise, training would halt after 100 epochs.

We implemented our neural architecture with the PyTorch library (Paszke et al., 2019). With a fixed random seed, model inputs were tokenized and truncated to the maximum sequence length of the selected Latin LM. They were grouped into batches of size 16 for all LMs save for CANINE-C and CANINE-S, as such models stressed memory resources with a maximum sequence length of 2048; in this case, we used a batch size of 8.

When Transformers were used, we fixed their attention heads to 8, used ReLU activations, and applied PreNorm (Chen et al., 2018; Nguyen and Salazar, 2019). We used either cross-entropy or GDW-CE to compute the loss. We optimized the neural networks with the Adam optimizer (Kingma and Ba, 2015), and gradients were clipped with an L_2 norm of 1 (Pascanu et al., 2013).

4.2. Hyperparameter Search

To avoid falling prey to poor hyperparameter selections for each instance of our architecture, we perform a random hyperparameter search (Bergstra and Bengio, 2012) of four trials for each instance. We vary the learning rate, hidden size, and number of layers in the encoder. We provide the ranges for these values with this work's repository.

Instances were constructed by fixing four modeling components: the embedding, the encoder, the dataset, and the loss function. SPhilBERTa was only employed with the PC dataset, as it was used to create the Gaussian dataset; moreover, it only used the identity-based encoder, as it creates a sequence-level embedding. Finally, the GDW-CE loss was only applied with the PC dataset.

Embedding	Encoder		
	Identity	LSTM	Transformer
Latin BERT	0.38	0.38	0.38
LaBERTa	0.31	0.31	0.37
PhilBERTa	0.24	**0.39**	**0.41**
mBERT	0.19	0.20	0.30
CANINE-C	0.26	0.33	0.24
CANINE-S	0.27	0.37	0.30

Table 3: *Odes* Macro-F_1 scores for models trained with data annotated with Gaussian clustering.

Once all sets of four trials were finished, we evaluated these models on the automatically-annotated test set. The best model among these four was then tested on the *Odes* data.

5. Results

We present a sampling of our experimental results in Tables 2 and 3, emboldening top two results across both tables. According to the *Odes* test set, the Gaussian dataset had a more reliable signal for sentiment. Our top two results used PhilBERTa embeddings with non-identity encoders. We submitted these models to the shared task, labeling the Transformer encoder model as our first submission and the BiLSTM encoder model as our second.

We provide our results in the shared task in Fig. 3. The first submission generally outperformed the second, only falling below the other on our worst-performing split: Seneca's *Hercules Furens*. When considering other teams' submissions, our first submission achieved the best macro-averaged Macro-F_1 score on the Pontano split by 0.1 points, and it narrowly missed tying for the top overall score (merited by TartuNLP) by 0.01 points. Thus, although our method did not place first, it nevertheless closely rivaled the best-performing method.

One question arising from our results concerns why the Gaussian dataset broadly outperformed the PC dataset. We speculate that this relates to the distributions of the underlying data, as presented in Table 1. The PC dataset heavily favored the neutral class; whether this resembles the true distribution or not, it poorly matched the distributions of the test set. The neutral class is consistently the smallest class among the emotionally-charged poems (Horace), lullabies (Pontano), and tragedy (Seneca) in the test set. Conversely, the Gaussian dataset has a more balanced spread of classes. Yet the lean of the Gaussian dataset's distribution into the positive class may help to explain our model's first-place performance on the Pontano subset.

To provide further evidence for this claim, we depict confusion matrices for our best-performing

Split	Macro-Avg. Score (↑)	 Rank (↓)	Micro-Avg. Score (↑)
Horace	0.29	3	–
Pontano	0.42	1	–
Seneca	0.12	4	–
Total	0.28	2	0.22

Split	Macro-Avg. Score (↑)	 Rank (↓)	Micro-Avg. Score (↑)
Horace	0.21	4	–
Pontano	0.31	3	–
Seneca	0.14	3	–
Total	0.22	4	0.22

Figure 3: Ranks and reported Macro-F_1 score averages for our EvaLatin 2024 shared task submissions. The left and right tables are for the first and second submissions, respectively. Ranks range between 1 and 4, not accounting for the baseline. When a tie occurs, the best possible ranking is displayed.

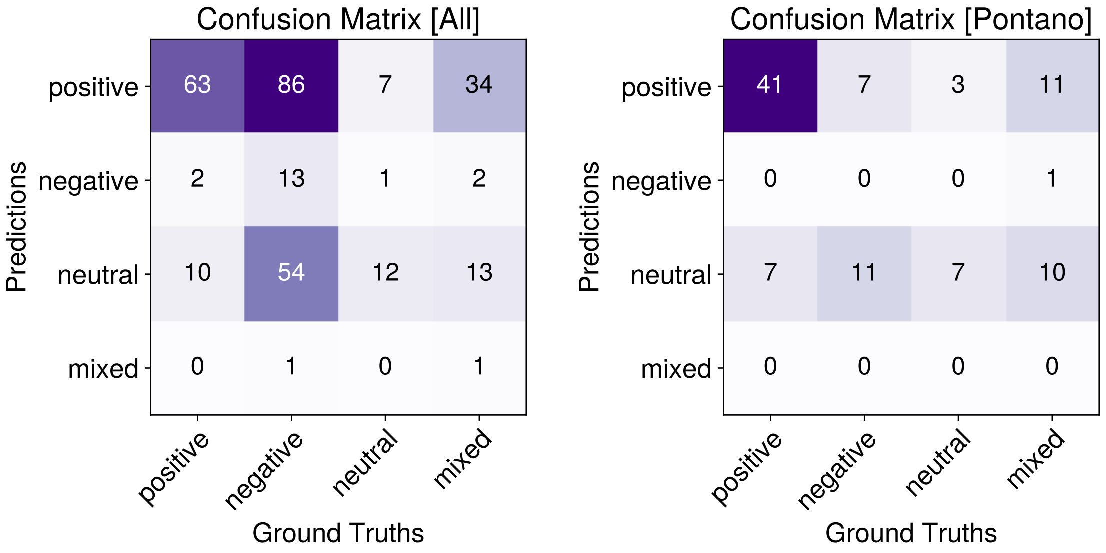


Figure 4: Confusion matrices for our best-performing submission. The left matrix is for the whole EvaLatin 2024 test set, whereas the right matrix is for the Pontano subset. Darker colors indicate larger values on the heatmap; text colors are shifted for readability.

submission in Figure 4.[3] For both the whole test set and the Pontano subset, the model primarily predicted the positive class, followed by the neutral class. In the case of the full dataset, these positive guesses add up to the largest sources of error: the model frequently mistakes negative sentences for positive ones. This effect is drastically reduced in the Pontano subset, as most of the sentences are positive. Altogether, these points further signal the meaningful influence of the Gaussian dataset's distribution on the model's performance.

To examine this influence in more detail, we check the level of agreement between our best neural models and the original Gaussian clustering annotator. Running EvaLatin's test data through the Gaussian model, we use Cohen's κ (Cohen, 1960; Artstein and Poesio, 2008) to measure our models' agreement beyond chance. Our top two neural models, which were trained on the Gaussian model's automatically annotated data, have κ values of 0.32 and 0.38. These weak agreement scores in combination with the prior evidence seem to imply that, although the neural models roughly inherited the Gaussian annotator's classification distribution, the networks' additional learning produced distinct cues for classification labels. Such effects may be ripe material for further investigation in improving low-resource polarity detection.

6. Conclusion

This paper presents two methods for data augmentation in a low-resource context. Each method employs a clustering-based approach to automatically annotate Latin data for polarity detection. The best of our models, using PhilBERTa-based embeddings, a Transformer encoder, and our dataset derived from Gaussian clustering, placed second in the task based on the macro-averaged Macro-F_1 score. Future work could explore the refinement of automatically-annotated data, perhaps integrating the expectation-maximization style of Gaussian training into a neural network to account for noise.

[3] The matrices for our other submission are quite similar, so the trends described also apply to it.

7. Acknowledgements

We would like to thank the EvaLatin organizers for their work in facilitating this shared task. We would also like to thank Margherita Fantoli for providing some clarifications about the Archimedes Latinus treebank (Fantoli and de Lhoneux, 2022). Finally, we would like to thank Darcey Riley, Ken Sible, Aarohi Srivastava, Chihiro Taguchi, Andy Yang, and Walter Scheirer for their feedback and support.

This research was supported in part by an FRSP grant from the University of Notre Dame.

8. Bibliographical References

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

David Bamman and Patrick J. Burns. 2020. Latin BERT: A contextual language model for classical philology.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. CANINE: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Graves and Juergen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4.

Kyle P. Johnson, Patrick J. Burns, John Stewart, Todd Cook, Clément Besnier, and William J. B. Mattingly. 2021. The Classical Language Toolkit: An NLP framework for pre-modern languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 20–29, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Toan Q. Nguyen and Julian Salazar. 2019. Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA. PMLR.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Frederick Riemenschneider and Anette Frank. 2023a. Exploring large language models for classical philology. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15181–15199, Toronto, Canada. Association for Computational Linguistics.

Frederick Riemenschneider and Anette Frank. 2023b. Graecia capta ferum victorem cepit. Detecting Latin allusions to Ancient Greek literature.

James A. Russell. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178.

James A. Russell and Albert Mehrabian. 1977. Evidence for a three-factor theory of emotions. *Journal of Research in Personality*, 11(3):273–294.

Rachele Sprugnoli, Federica Iurescia, and Marco Passarotti. 2024. Overview of the EvaLatin 2024 evaluation campaign. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages – LT4HALA 2024*, Torino, Italy. European Language Resources Association.

Leimin Tian, Catherine Lai, and Johanna Moore. 2018. Polarity and intensity: The two aspects of sentiment analysis. In *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*, pages 40–47, Melbourne, Australia. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272.

9. Language Resource References

David Bamman and Gregory Crane. 2011. The Ancient Greek and Latin dependency treebanks. In *Language Technology for Cultural Heritage*, Theory and Applications of Natural Language Processing, pages 79–98. Springer Berlin Heidelberg, Berlin, Heidelberg.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

Flavio Massimiliano Cecchini, Timo Korkiakangas, and Marco Passarotti. 2020a. A new Latin treebank for Universal Dependencies: Charters between Ancient Latin and Romance languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 933–942, Marseille, France. European Language Resources Association.

Flavio Massimiliano Cecchini, Rachele Sprugnoli, Giovanni Moretti, and Marco Passarotti. 2020b. UDante: First steps towards the Universal Dependencies treebank of Dante's Latin works. In *Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020, Bologna, Italy, March 1-3, 2021*, volume 2769 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Yanqing Chen and Steven Skiena. 2014. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 383–389, Baltimore, Maryland. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Margherita Fantoli and Miryam de Lhoneux. 2022. Linguistic annotation of neo-Latin mathematical texts: A pilot-study to improve the automatic parsing of the Archimedes Latinus. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 129–134, Marseille, France. European Language Resources Association.

Dag T. Haug, Marius L. Jøhndal, Hanne M. Eckhoff, Eirik Welo, Mari J. B. Hertzenberg, and Angelika Müth. 2009. Computational and linguistic issues in designing a syntactically annotated parallel corpus of Indo-European languages. In *Traitement Automatique Des Langues, Volume 50, Numéro 2: Langues Anciennes [Ancient Languages]*, pages 17–45, France. ATALA (Association pour le Traitement Automatique des Langues).

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Caitlin A. Marley. 2018. *Sentiments, Networks, Literary Biography: Towards a Mesoanalysis of Cicero's Corpus*. Ph.D. thesis, University of Iowa.

Marco Passarotti. 2019. The Project of the Index Thomisticus Treebank. In *Digital Classical Philology*, volume 10, pages 299–320. De Gruyter, Berlin, Boston.

David A. Smith, Jeffrey A. Rydberg-Cox, and Gregory R. Crane. 2000. The Perseus project: A digital library for the humanities. *Literary and linguistic computing*, 15(1):15–25.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Rachele Sprugnoli, Francesco Mambrini, Marco Passarotti, and Giovanni Moretti. 2023. The sentiment of Latin poetry. Annotation and automatic analysis of the Odes of Horace. *IJCoL. Italian Journal of Computational Linguistics*, 9(1):53–71.

Rachele Sprugnoli, Marco Passarotti, Flavio Massimiliano Cecchini, Margherita Fantoli, and Giovanni Moretti. 2022. Overview of the EvaLatin 2022 evaluation campaign. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 183–188, Marseille, France. European Language Resources Association.

Rachele Sprugnoli, Marco Passarotti, Daniela Corbetta, and Andrea Peverelli. 2020. Odi et amo. Creating, evaluating and extending sentiment lexicons for Latin. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3078–3086, Marseille, France. European Language Resources Association.

TartuNLP at EvaLatin 2024: Emotion Polarity Detection

Aleksei Dorkin, Kairit Sirts
University of Tartu
Tartu, Estonia
{aleksei.dorkin, kairit.sirts}@ut.ee

Abstract

This paper presents the TartuNLP team submission to EvaLatin 2024 shared task of the emotion polarity detection for historical Latin texts. Our system relies on two distinct approaches to annotating training data for supervised learning: 1) creating heuristics-based labels by adopting the polarity lexicon provided by the organizers and 2) generating labels with GPT4. We employed parameter efficient fine-tuning using the adapters framework and experimented with both monolingual and cross-lingual knowledge transfer for training language and task adapters. Our submission with the LLM-generated labels achieved the overall first place in the emotion polarity detection task. Our results show that LLM-based annotations show promising results on texts in Latin.

Keywords: emotion polarity classification, adapter training, knowledge transfer, latin

1. Introduction

This short report describes the system developed the TartuNLP team for the Emotion Polarity Detection task of the EvaLatin 2024 Evaluation Campaign (Sprugnoli et al., 2024). The goal of the task was to label Latin texts from three historical authors with four emotion polarity labels as positive, negative, neutral or mixed. For this task, no training data was provided, but only a polarity lexicon and a small evaluation set with 44 annotated sentences.

Our approach entails two steps. First, we annotated data for supervised model training a) via heuristic rules using the provided polarity lexicon and b) using GPT-4 (see Section 2). Secondly, we adopted knowledge transfer with parameter-efficient training via adapters (Houlsby et al., 2019) followed by task-specific fine-tuning on the data annotated in the first step (see Section 3). The knowledge transfer was applied both cross-lingually via pretraining on an English sentiment analysis task, and monolingually by training on an unannotated Latin text corpus.

We made two submissions to the shared task: one with heuristically annotated training data and another with the GPT-4 annotated labels. Both submissions obtained competitive results, with the submission with GPT-4 labels obtaining the first place overall. The code for the system is available on GitHub.[1]

2. Data Annotation

For the Emotion Polarity Detection task, no training data was provided. However, the organizers provided two useful resources: a polarity lexicon and a small gold annotated sample. We employed two distinct approaches to annotate the training data based on these resources: a heuristics-based and an LLM-based. The annotated data from both approaches is available on HuggingFace Hub.[2] The label distribution for the annotated data is presented in Table 1.

Label	Heuristics	LLM-based
positive	6535	1334
negative	2243	1028
mixed	5884	221
neutral	735	4698
Total	15396	7281

Table 1: Statistics of the annotated training data.

2.1. Heuristics-based annotation

In this approach, we employed the provided polarity lexicon similarly to the lexicon-based classifier by Sprugnoli et al. (2023). First, data from all available Universal Dependencies (Zeman et al., 2023) sources (Version 2.13, the most recent one at the time of writing) in Latin was collected :

1) Index Thomisticus Treebank (ITTB);
2) Late Latin Charter Treebank (LLCT);
3) UDante;
4) Perseus;
5) PROIEL treebank.

Then, the sentences containing no nouns or adjectives in the lexicon were removed. The filtered sentences were assigned labels based on the following rules:

[1] https://github.com/slowwavesleep/ancient-lang-adapters/tree/lt4hala

[2] https://huggingface.co/datasets/adorkin/evalatin2024

1) If all words in the sentence are neutral according to the polarity lexicon, the sentence was labeled as neutral;
2) If the mean polarity of the words in the sentence is in the range from -0.1 to 0.1, then the sentence was labeled as mixed;
3) If the mean polarity is larger than 0.1, then the sentence was labeled as positive;
4) If the mean polarity is less than 0.1, then the sentence was labeled as negative.

Our expectation from this approach was that training a model on lexicon-annotated data would result in a model with better generalization capabilities than simply applying the lexicon classifier. The total amount of sentences annotated this way was 15396.

2.2. LLM-based annotation

In this approach, we made use of the OpenAI's GPT-4 model via the API (`gpt-4-turbo-preview`[3]). The sentences were again sampled from the Universal Dependencies sources. The model was given the description of the problem and one example per label from the gold annotations file. The model was tasked with assigning the given sentence a label and providing an explanation as to why it assigned that particular label.

With this approach, we expected that GPT-4 could simulate the annotation process done by an expert in Latin. According to the first author's somewhat limited understanding of Latin and based on a small sample of annotations and explanations done by the model, the output seems reasonable. We set out to spend about 15 euros per data annotation, which after removing sentences with invalid labels resulted in 7281 annotated sentences.

3. Description of the system

The system in our submission is based on the BERT architecture (Devlin et al., 2019). More specifically, we employed the multilingual version of RoBERTa (Zhuang et al., 2021)—XLM-RoBERTa (Conneau et al., 2020), which was trained on the data that included Latin.

We treated Emotion Polarity Detection as a multiclass classification problem and fine-tuned the model accordingly. However, instead of full fine-tuning, we trained a stack of adapters: a language adapter and a task adapter. Training adapters involves adding a small number of trainable parameters to the model while freezing the rest of the parameters (Houlsby et al., 2019). In addition to making the training considerably faster, adapters mitigate overfitting and catastrophic forgetting, which are common problems when dealing with small amounts of training data. We implemented our system by using the transformers[4] and the adapters[5] libraries.

We expected the model to benefit from both mono-lingual and cross-lingual knowledge transfer; therefore, the training process comprised several stages. First, we fine-tuned a Latin language adapter on a publicly available Latin Corpus[6] collected from the Latin Library[7]. In the next phase of training, we trained a task-specific classification adapter on the English IMDB movie reviews dataset[8]. The dataset contains only two labels: positive and negative. We created an adapter with a classification head with four classes, two of which remained unused during this stage. Finally, we stacked the task adapter previously trained on English on top of the language adapter, and continued training the task adapter on the annotated data in Latin.

The language adapter was trained for ten epochs with a learning rate 1e-4. For further usage, we took the last checkpoint. The task adapter was trained on data in English for five epochs with a learning rate of 5e-4, and we also took the last checkpoint. Finally, for the submissions, we trained a model on both sets of annotated data for 50 epochs with a 5e-4 learning rate. We used the provided gold annotation example as the validation set for training and measured the F-score on it after each epoch. For submission, we selected the best checkpoint based on the validation F-score.

4. Results

We made two submissions to the Emotion Polarity Detection task; the first one (TartuNLP_1) fine-tuned on the dataset with the heuristic labels, and the second one (TartuNLP_2) fine-tuned on the dataset with the LLM-generated labels. Both submissions obtained competitive results, with the model trained on the LLM-annotated labels (TartuNLP_2) taking the overall first place and the model trained on the heuristics-annotated data (TartuNLP_1) taking the second place on micro average F1-score and the third place on the macro average F1-score (see Table 2).

While the scores obtained by the two models are quite close, there is frequent disagreement in their predictions: out of 294 test examples, the models

[3] https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo

[4] https://github.com/huggingface/transformers

[5] https://github.com/adapter-hub/adapters

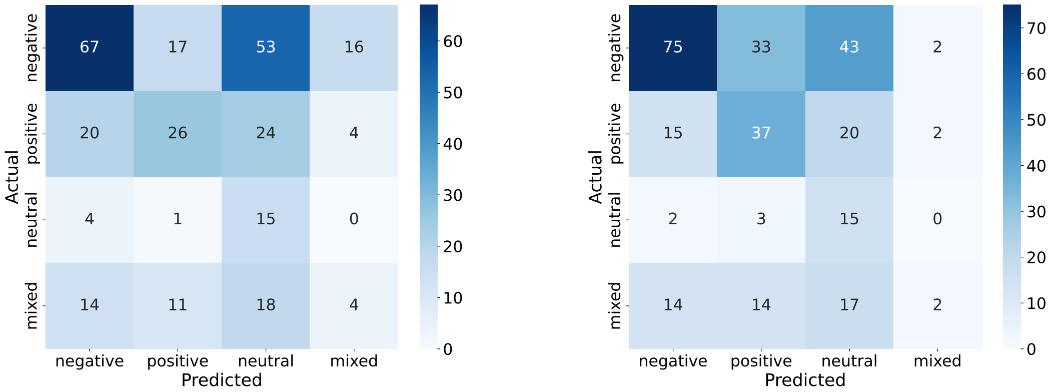
[6] https://github.com/mathisve/LatinTextDataset

[7] https://www.thelatinlibrary.com/

[8] https://huggingface.co/datasets/imdb

Model	Micro Average F1	Macro Average F1
TartuNLP_2	**0.34**	**0.29**
TartuNLP_1	0.32	0.27
NostraDomina_1	0.22	0.28
NostraDomina_2	0.22	0.22

Table 2: The overall results of all teams.



(a) TartuNLP_1 with lexicon-based heuristic labels.

(b) TartuNLP_2 with GPT4-generated labels.

Figure 1: Confusion matrices for both submissions.

disagreed in 140 examples. In case of disagreement, the heuristics- and LLM-based models made correct predictions in 40 and 57 examples respectively. Meanwhile, in case of agreement, the models correctly predicted the labels of 72 examples out of 154.

The confusion matrices for both models (see Figure 1) are similar. The models had the most trouble with the mixed class, while the negative class was the easiest to predict; this is in line with findings by Sprugnoli et al. (2023), who reported the lowest inter-annotator agreement for the mixed class, while the negative class had the highest agreement, assuming that the test data of the shared task was annotated in a similar manner.

We performed a small ablation study on the labeled test data released by the organizers after evaluating the shared task results to measure the effect of the knowledge transfer methods used:

1) Monolingual knowledge transfer from the wider Latin corpus in training the language adapter;
2) Cross-lingual knowledge transfer from the English IMDB sentiment dataset in training the task adapter.

The results of the study, shown in Table 3, were somewhat unexpected. First of all, we observe that the base model with no knowledge transfer is already as good or better than the submitted models adopting both types of knowledge transfer. Secondly, the monolingual knowledge transfer by training the language adapter improves the micro-averaged F1-score with both types of labels. Finally, the model with the LLM-generated labels benefits more from the monolingual language adapter training resulting in a model that noticeably outperforms our initial submission.

5. Discussion

The model with LLM-generated labels obtained better results than the model with lexicon-based heuristic labels, although the final results of both submitted systems are relatively close. However, the ablation study testing the effectiveness of both monolingual and cross-lingual knowledge transfer demonstrated that the model trained on the LLM-annotated data can show even better results when omitting the cross-lingual transfer from English. This is despite the fact that the number of LLM-annotated examples was nearly twice as small, suggesting that the LLM annotations are of higher quality than the labels based on lexicon-informed heuristics.

Despite our model trained on the LLM-annotated data taking the overall first place, the absolute values are somewhat low and sometimes below the baseline. There might be several reasons related to the choice of the data source and the annotation scheme and procedures. First, many of the exam-

Ablation	**Micro Avg F1**	**Macro Avg F1**	**Val F1**
Heuristic labels without knowledge transfer	0.33	0.26	0.48
Heuristic labels + Monolingual language transfer	0.34	0.25	0.48
Heuristic labels + Cross-lingual task transfer	0.30	0.23	0.55
Heuristic labels + Both (TartuNLP_1)	0.32	0.27	0.47
LLM labels without knowledge transfer	0.37	0.30	0.55
LLM labels + Monolingual language transfer	**0.38**	**0.30**	**0.61**
LLM labels + Cross-lingual task transfer	0.37	0.29	0.53
LLM labels + Both (TartuNLP_2)	0.34	0.29	0.48

Table 3: The results of the ablation study.

ples appear to be expository or narrative in nature. It is difficult to assign a particular emotive polarity to the texts of that kind. Furthermore, Sprugnoli et al. (2023) mention that the annotators were instructed to assign labels on the sentence level. However, they were also presented with the wider context of the sentence. This leads us to believe that some labels are actually contextual, especially when the annotated sentence contains only a single word (for example, the sentence "Mentior?" is labeled as mixed). Secondly, the manual analysis of the examples shows that it is quite difficult to distinguish between mixed and neutral texts. This appears to be true for the trained models, as well.

One possibility of improvement is to reframe the task as a multi-label classification problem instead. The model would be expected to predict the probabilities for the negative and positive labels independently. If the probability of both labels is low, the assigned label can be "neutral"; if both probabilities are high, the label can be "mixed"; otherwise, the label corresponding to the highest probability would be assigned.

6. Conclusion

This paper described our solution to the Emotion Polarity Detection task of the EvalLatin Evaluation Campaign. Our submission obtained with a model trained on a dataset with LLM-generated labels achieved the overall first place, showing that LLM-based annotations can be useful for processing texts in Latin.

7. Bibliographical References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Rachele Sprugnoli, Federica Iurescia, and Marco Passarotti. 2024. Overview of the EvaLatin 2024 evaluation campaign. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages â€“ LT4HALA 2024*, Torino, Italy. European Language Resources Association.

Rachele Sprugnoli, Francesco Mambrini, Marco Passarotti, and Giovanni Moretti. 2023. The sentiment of latin poetry. annotation and automatic analysis of the odes of horace. *IJCoL. Italian Journal of Computational Linguistics*, 9(9-1).

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Salih Furkan Akkurt, Gabrielė Aleksandravičiūtė, Ika Alfina, Avner Algom, Khalid Alnajjar, Chiara Alzetta, Erik Andersen, Lene Antonsen, Tatsuya Aoyama, Katya Aplonova, Angelina Aquino, Carolina Aragon, Glyd Aranes, Maria Jesus Aranzabe, Bilge Nas Arıcan, Ĥórunn Arnardóttir, Gashaw

Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Katla Ásgeirsdóttir, Deniz Baran Aslan, Cengiz Asmazoğlu, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Mariana Avelãs, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Shabnam Behzad, Kepa Bengoetxea, İbrahim Benli, Yifat Ben Moshe, Gözde Berk, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, António Branco, Kristina Brokaitė, Aljoscha Burchardt, Marisa Campos, Marie Candito, Bernard Caron, Gauthier Caron, Catarina Carvalheiro, Rita Carvalho, Lauren Cassidy, Maria Clara Castro, Sérgio Castro, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Liyanage Chamila, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Daniela Corbetta, Francisco Costa, Marine Courtin, Mihaela Cristescu, Ingerid Løyning Dale, Philemon Daniel, Elizabeth Davidson, Leonel Figueiredo de Alencar, Mathieu Dehouck, Martina de Laurentiis, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Adrian Doyle, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Christian Ebert, Hanne Eckhoff, Masaki Eguchi, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Farah Essaidi, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Federica Favero, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Amal Fethi, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Federica Gamba, Marcos Garcia, Moa Gärdenfors, Fabrício Ferraz Gerardi, Kim Gerdes, Luke Gessler, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Takahiro Harada, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Marivel Huerta Mendez, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Artan Islamaj, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Katharine Jiang, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Ritván Karahóğa, Andre Kåsen, Tolga Kayadelen, Sarveswaran Kengatharaiyer, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Alexey Koshevoy, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Adrian Kuqi, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Kris Kyle, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Lauren Levine, Cheuk Ying Li, Josie Li, Keying Li, Yixuan Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Yi-Ju Jessica Lin, Krister Lindén, Yang Janet Liu, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Ilya Makarchuk, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Stella Markantonatou, Héctor Martínez Alonso, Lorena Martín Rodríguez, André Martins, Cláudia Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Aaron Miller, Karina Mischenkova, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Hulda Óladóttir,

Adédayọ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Noam Ordan, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Teresa Paccosi, Alessio Palmero Aprosio, Anastasia Panova, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Giulia Pedonese, Angelika Peljak-Łapińska, Siyao Peng, Siyao Logan Peng, Rita Pereira, Sílvia Pereira, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Andrea Peverelli, Jason Phelan, Jussi Piitulainen, Yuval Pinter, Clara Pinto, Tommi A Pirinen, Emily Pitler, Magdalena Plamada, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Robert Pugh, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andreia Querido, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Joana Ramos, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Arij Riabi, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Ivan Roksandic, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Ben Rozonoyer, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Marta Sartor, Mitsuya Sasaki, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Maria Shvedova, Janine Siewert, Einar Freyr Sigurðsson, João Silva, Aline Silveira, Natalia Silveira, Sara Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Haukur Barri Símonarson, Kiril Simov, Dmitri Sitchinava, Ted Sither, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Per Erik Solberg, Barbara Sonnenhauser, Shafi Sourov, Rachele Sprugnoli, Vivian Stamou, Steinĥór Steingrímsson, Antonio Stella, Abishek Stephen, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Daniel Swanson, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Mirko Tavoni, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Sara Tonelli, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sveinbjörn Ĥórðarson, Vilhjálmur Ĥorsteinsson, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Elena Vagnoni, Sowmya Vajjala, Socrates Vak, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Uliana Vedenina, Giulia Venturi, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Shira Wigderson, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Taksum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Yilun Zhu, Anna Zhuravleva, and Rayan Ziane. 2023. Universal dependencies 2.12. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A Robustly Optimized BERT Pre-training Approach with Post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

Overview of EvaHan2024: The First International Evaluation on Ancient Chinese Sentence Segmentation and Punctuation

Bin Li[1, 2, 3] Bolin Chang[1, 2] Zhixing Xu[1, 2] Minxuan Feng[1, 2] Chao Xu[1, 2]
Weiguang Qu[4, 2] Si Shen[5] Dongbo Wang[6, 2]
1. School of Chinese Language and Literature, Nanjing Normal University, China
2. Center for Language Big Data and Computational Humanities, Nanjing Normal University, China
3. Faculty of Arts and Humanities, University of Macau, China
4. School of Computer and Electronic Information, Nanjing Normal University, China
5. School of Economics and Management, Nanjing University of Science and Technology, China
6. College of Information Management, Nanjing Agricultural University, China
E-mail: db.wang@njau.edu.cn

Abstract

Ancient Chinese texts have no sentence boundaries and punctuation. Adding modern Chinese punctuation to theses texts requires expertise, time and efforts. Automatic sentence segmentation and punctuation is considered as a basic task for Ancient Chinese processing, but there is no shared task to evaluate the performances of different systems. This paper presents the results of the first ancient Chinese sentence segmentation and punctuation bakeoff, which is held at the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) 2024. The contest uses metrics for detailed evaluations of 4 genres of unpublished texts with 11 punctuation types. Six teams submitted 32 running results. In the closed modality, the participants are only allowed to use the training data, the highest obtained F1 scores are respectively 88.47% and 75.29% in sentence segmentation and sentence punctuation. The perfermances on the unseen data is 10 percent lower than the published common data, which means there is still space for further improvement. The large language models outperform the traditional models, but LLM changes the original characters around 1-2%, due to over-generation. Thus, post-processing is needed to keep the text consistancy.

Keywords: Ancient Chinese, Sentence Segmentation, Sentence Punctuation, Evaluation

1. Introduction

The EvaHan series represents an international endeavor focusing on the advancement of information processing for ancient Chinese texts. In 2022, EvaHan was convened in Marseille, France, where it conducted evaluations on word segmentation and part-of-speech tagging in ancient Chinese, contributing to the field’s understanding of these fundamental tasks (Li et al., 2022). The following year, the series moved to Macau, China, extending its scope to include evaluations on ancient Chinese machine translation, a significant step in computational linguistics for historical languages (Wang et al., 2023). In 2024, EvaHan is set to pioneer a new frontier with its first campaign specifically devoted to the evaluation of Ancient Chinese Sentence Segmentation and Punctuation, aiming to address a critical and yet under-explored area in the processing of classical texts.

In the natural language processing (NLP) tasks like speech to text recognition and chat text punctuation, texts often lack correct or appropriate sentence boundaries and punctuation (Nagy et al., 2021), a situation that increases the complexity of processing and reduces efficiency (Jones et al., 2003; Tündik et al., 2018). To enhance subsequent task processing, it is essential to add correct sentence boundaries and punctuation to these texts (Peitz et al., 2011). Addressing this, recent research has explored using large language models for automatically punctuating text in tasks such as text analysis and speech processing (Kolár and Lamel, 2012; González-Docasal et al., 2021; Bakare et al., 2023). Given the critical role of punctuation in text interpretation, comprehensive evaluations have been conducted to assess the effectiveness of automatic punctuation in NLP tasks (Meister et al., 2023). These evaluations have developed scientific indicators for texts in English and other languages, forming a complete and robust evaluation system.

Ancient Chinese also has no sentence boundaries and punctuation, making it quite hard to read (Lyu et al., 1983). Nowadays, in most republished ancient Chinese books punctuation is added manually by language experts. Here is an example of ancient Chinese.

(1) 亟 請 於 武公 公 弗 許
repeatedly request to Wugong Wugong not accept
(Wu Jiang) repeatedly requested Wugong, but he refused.

Table 1 shows the sentence boundaries and punctuation added to Exp 1.

Raw Text	亟請於武公公弗許
+Sentence Segmentation	亟請於武公 公弗許
+Sentence Punctuation	亟請於武公，公弗許。

Table 1: Example of adding sentence segmentation and punctuation.

With the establishment of the modern Chinese punctuation system, important texts of ancient books republished nowadays all include punctuation, which are much easier to read. But this work requires experts with great language knowledge of ancient Chinese. For example, a scholar usually needs several months to finish one book with around 200,000 characters. The great costs of time, funds and efforts place constraints on republication of these texts. And there is still a huge number of ancient books need to be processed. But most ancient books do not have that great value to be republished in paper books. The electronic texts could be automatically processed for many NLP tasks and applications, such as knowledge mining, Q&A, and machine translation (Sommerschield et al., 2023).

Therefore, automatic sentence segmentation and punctuation in ancient Chinese are fundamental tasks for compiling and publishing ancient books as well as ancient Chinese information processing, laying the foundation for subsequent tasks (Su et al., 2021). In recent years, research on sentence segmentation and punctuation in ancient Chinese have achieved good results (Chen et al., 2007; Huang et al., 2008; Shi et al., 2019; Yu et al., 2019; Cheng et al., 2020; Hong et al., 2021; Hu et al., 2021; Yuan et al., 2022), yet encountering some challenges.

Firstly, the number of types of punctuation used in existed automatic annotation systems vary from the basic 4 punctuation to 8. As a result, it is not easy to judge or compare the performances of the systems. Secondly, sentence segmentation and punctuation are usually conducted in a pipeline. Sentence segmentation errors will easily spread to the punctuation process. Thirdly, the evaluation paradigm for sentence segmentation and punctuation were not fully set up. The data set used for sentence segmentation and punctuation was disorganized, potentially due to the integration of test sets with training sets in the pre-training of large language models. And in calculating model scores, most studies rely on character-based assessments rather than punctuation-based assessments. Sentence segmentation and punctuation in ancient Chinese necessitate an evaluation task to address these challenges, to standardize irregular processes, and to provide a benchmark.

EvaHan2024 aims to give a good evaluation metric for this joint task and to answer three main questions:

- How can modern punctuation be integrated into ancient texts that lack sentence boundary and punctuation?
- Could the methodology of large language models facilitate processing ancient Chinese information?
- To ensure the integrity of the evaluation process, particularly given that large language models are trained on extensive collections of ancient Chinese texts, what strategies can be employed to prevent the overlap of the test corpus with the training set?

EvaHan2024 is proposed as part of the Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA), co-located with LREC 2024. Scorer and detailed guidelines are all available in our GitHub repository[1].

2. Task

EvaHan2024 consolidated the following two problems into a joint task:

- Sentence segmentation involves converting Chinese text into a sequence of sentences, with each sentence separated by a single space.
- Sentence punctuation, on the other hand, focuses on the accurate placement of appropriate punctuation marks at the end of each sentence to ensure clarity.

In this shared task, a sentence should be automatically parsed from raw text to punctuated text shown in Table 1. There are eleven types of punctuation involved in the evaluation, as shown in Table 2. The evaluation toolkit gives the scores on both sentence segmentation and punctuation. EvaHan2024 does not accept running results with sentence segmentation only.

Punctuation	Name
，	Comma
。	Period
、	Slight-pause
：	Colon
；	Semicolon
？	Question Mark
！	Exclamation Mark
“	Left Quote
”	Right Quote
《	Left Book Title Mark
》	Right Book Title Mark

Table 2: 11 Punctuation involved in the evaluation

3. Dataset

The training dataset of EvaHan2024 is extracted from the classic historical books *Siku Quanshu* (四库全书)[2], the test data is extracted from 4 unpublished books. The comparison dataset is the text from *Zuozhuan*[3]. All the data has been punctuated and proofread by experts of Ancient Chinese language.

3.1 Data Format

The dataset consists of two parts, a training dataset and two test datasets, as shown in Table 3. All the punctuation are annotated by following *General Rules for Punctuation* (2012) and *Academic Publishing Specification-Collation of Chinese Ancient Books* (2015). All texts are encoded in UTF-8 plain text files.

[1] https://circse.github.io/LT4HALA/2024/EvaHan

[2] https://en.wikipedia.org/wiki/Siku_Quanshu

[3] https://catalog.ldc.upenn.edu/LDC2017T14

As there are no sentence boundaries in Chinese texts, the raw texts only contain Chinese characters. After manual annotation, sentence punctuation are added to the text. As shown in Table 1, each sentence is marked with punctuation.

Data Sets	Sources	# Char Tokens	# Punctuation Tokens
Train	*Siku Quanshu*	19,796,102	3,929,523
TestA	4 genres of texts	50,306	9,673
TestB	*Zuozhuan*	196,560	53,919

Table 3: Texts distributed as training/test data in EvaHan2024.

3.2 Training Data

The training data includes punctuated text sourced from *Siku Quanshu* (四库全书), the largest series of ancient Chinese books , assembled during the Qing Dynasty. *Siku Quanshu* comprises four volumes including Jing, Shi, Zi and Ji, approximately 997 million words in total.

3.3 Test Data

Test Data was supplied in its raw format, consisting of Chinese characters only. Gold data was released after the evaluation period.

There are two test datasets. Blind *TestA* is designed to see how a system performs on dissimilar data. *TestA* includes four genres, namely *Products in Local Chronicles* (方志物产), *County Annals* (县志), *Buddhist Sutra* (佛经) and *Academy Records* (书院志), as shown in Table 4. TestA was not publicaly released/published publicly before EvaHan. This is an important way to ensure that no test data has been used by training procedure, especially for the LLM pre-training.

Genres	# Char Tokens	# Punctuation Tokens
Products in Local Chronicles (方志物产)	6,578	1,982
County Annals (县志)	24,548	4,244
Buddhist Sutra (佛经)	9,854	1,957
Academy Records (书院志)	9,326	1,490

Table 4: Four genres of TestA

We also compiled up a comparison test set *TestB*, which is designed to see how a system performs on similar data from the training data. *TestB* is the text of *Zuozhuan (左传)*, an ancient Chinese work believed to date back to the Warring States Period (475-221 BC). Specifically, *Zuozhuan* is a commentary on the *Chunqui (春秋)*, a history of the Chinese Spring and Autumn period (770-476 BC). *TestB* is partially included in the training set, and it can be easily obtained from the web. But the teams are not allowed to use it as training data directly. There have been several papers reporting their performance on this data (Shi et al., 2010; Cheng 2020 et al., 2020). Its size is larger than testA, containing 196,560 characters and 53,919 punctuation.

As *Zuozhuan* is included in *Siku Quanshu*, utilized for pre-training large language models, *TestB* serves solely as a reference for comparison.

4. Evaluation

Initially, each team could only access the training data. Later, the unlabeled test data was released. After the submission, the labels for the test data was also released.

4.1 Scoring

The scorer employed for EvaHan is a modified version of the one developed from SIGHAN2008 (Jin and Chen, 2008). The evaluation aligned the system-produced sentences to the gold standard ones. Then, the performance of sentence segmentation and punctuation were evaluated by precision, recall and F1 score. In the scoring process, we assess the correctness of punctuation directly, rather than Chinese characters as done in previous researches. The final ranking was based on F1 score of auto punctuation.

4.2 Two Modalities

Each participant can submit runs following two modalities. In the closed modality, the resources each team could use are limited. Each team can only use the Training data *Train*, and *XunziALLM*[4], a large language model pretrained on a very large corpus of traditional Chinese collection, including *Siku Quanshu* (四库全书). Other resources are not allowed in the closed modality.

In the open modality, there is no limit on the resources, data and models. Annotated external data, such as the components or Pinyin of the Chinese characters, word embeddings can be employed, as shown in Table 5. But each team has to state all the resources, data and models they use in each system in the final report.

Limits	Closed Modality	Open Modality
Machine learning algorithm	No limit	No limit
Pretrained model	Only XunziALLM	No limit
Training data	Only *Train*	No limit
Features used	Only from *Train*	No limit
Manual correction	Not allowed	Not allowed

Table 5: Limitations on the two modalities.

4.3 Procedure

Training data was released for download from January 20, 2024. Test data was released on March

[4] https://github.com/Xunzi-LLM-of-Chinese-classics/XunziALLM

1, 2024, and results were due on 00:00 (UTC) March 8, 2024.

5. Participants and Results

5.1 Participants

A total of 17 teams registered for the task, with 6 of those teams ultimately submitting 32 entries. Table 6 presents the details of the participating teams. Notably, the majority of submissions were under the 'closed modality', with only one team opting for the 'open modality'. It is important to mention that 27 submissions were initially presented in incorrect formats, as indicated by the '+' symbol in Table 6. This issue, primarily attributed to the over-generation of language by large language models (LLM), was subsequently rectified by us to facilitate accurate evaluation.

ID	Name	Affiliation	TestA		TestB	
			C	O	C	O
1	BNU	Beijing Normal University	1^+	0	1^+	0
2	CT	China Telecom Corporation Ltd. AI Technology Company	1^+	1^+	1^+	1^+
3	MiDU	Beijing Midu Information Technology Co., Ltd.	7^+	0	7^+	0
4	NJU1	Nanjing University	1^+	0	3	0
5	NJU2	Nanjing University	1^+	0	1	0
6	SU	Soochow University	1^+	0	1	0

Table 6: Participating teams by Corpus and Modality (**C**losed and **O**pen). Files with “+” means that the LLM changes the original texts.

5.2 Results

Table 7-10 list the performance of the participating teams, arranged in descending order of the F1 scores for the sentence punctuation. The Precision, Recall and F1 score for Sentence Segmentation, are abbreviated as P_{seg}, R_{seg} and F_{seg}, respectively. Simliarly, for sentence punctuation, they are abbreviated as P_{punc}, R_{punc} and F_{punc}. We categorized the results submitted by the participants as *TestA* Closed, *TestA* Open, *TestB* Closed, and *TestB* Open. The results are ranked by the sentence punctuation scores. Most teams participated in the closed tests. It can be seen from the four tables that there is a high correlation between sentence segmentation and sentence punctuation.

For *TestA*, the highest F1 score of sentence punctuation is 75.29% in the closed modality. In the open modality, it is 72.12%.

The scores of sentence segmentation are much higher. CT scores 88.86% and 87.93% in the closed and open modality. It is remarkble that MiDU scores 88.47% in the closed modality, with a slightly higher score 75.29% for sentence punctuation.

For TestB, which is designed to see how the systems perform on similar data as the training set, the scores have all increased by approximately 5 to 10 points. NJU2 scores 82.43% in TestB, ranking the first place in the close modality. But they submit no result in the open modality, and this score is even higher than their performance on TestA.

Team	P_{seg}	R_{seg}	F_{seg}	P_{punc}	R_{punc}	F_{punc}
MiDU	91.05	86.04	88.47	78.81	72.07	**75.29**
SU	89.84	84.70	87.19	75.88	69.71	72.67
CT	91.11	86.72	**88.86**	74.34	68.49	71.30
NJU2	90.80	76.34	82.94	77.75	63.85	70.12
NJU1	90.93	75.57	82.54	74.15	60.14	66.41
BNU	90.93	71.61	80.12	73.83	56.92	64.28

Table 7: TestA closed modality (%)

Team	P_{seg}	R_{seg}	F_{seg}	P_{punc}	R_{punc}	F_{punc}
CT	90.78	85.24	87.93	75.64	68.92	72.12

Table 8: TestA open modality (%)

Team	P_{seg}	R_{seg}	F_{seg}	P_{punc}	R_{punc}	F_{punc}
NJU2	95.98	90.54	93.18	85.08	79.93	**82.43**
CT	96.32	91.46	**93.83**	85.99	79.10	82.40
SU	94.64	91.93	93.27	82.93	78.96	80.89
MiDU	95.05	90.05	92.48	82.92	77.30	80.01
NJU1	95.38	89.68	92.44	80.44	75.67	77.98
BNU	95.25	88.15	91.57	79.06	73.66	76.26

Table 9: TestB (for comparison only) in closed modality (%)

Team	P_{seg}	R_{seg}	F_{seg}	P_{punc}	R_{punc}	F_{punc}
CT	94.73	89.21	91.89	82.91	74.94	78.73

Table 10: TestB (for comparison only) in open modality (%)

5.3 Baselines

To provide a basis for comparison, we computed the baseline scores for each of the test sets.

5.3.1 Sentence Segmentation

The baseline for ancient Chinese sentence segmentation was constructed by *XunziALLM* (Xunzi-Qianwen-7B-CHAT) model, as shown in Table 11.

Testing Set	P_{seg}	R_{seg}	F_{seg}
TestA	90.53	66.12	76.42
TestB	95.28	87.17	91.04

Table 11: Sentence segmentation baselines (%)

The sentence segmentation scores of most teams exceed the baselines in TestA and TestB. The best scores outperform the baselines by around 10 points as shown in Table 12.

Testing Set	P_{seg}	R_{seg}	F_{seg}
TestA	91.11(+0.58)	86.72(+20.6)	88.86(+12.44)
TestB	96.32(+1.04)	91.46(+4.76)	93.83(+2.79)

Table 12: The promotion to the baselines of sentence segmentation (%)

5.3.2 Sentence Punctuation

The baseline for ancient Chinese sentence segmentation was constructed by *XunziALLM* model, as shown in Table 13.

Testing Set	P_{punc}	R_{punc}	F_{punc}
TestA	73.52	52.22	61.06
TestB	79.25	72.09	75.50

Table 13: Sentence punctuation baselines (%)

The sentence punctuation scores of most teams exceed the baselines in TestA and TestB. The best scores outperform the baselines by around 10 points as shown in Table 14.

Testing Set	P_{punc}	R_{punc}	F_{punc}
TestA	78.81(+5.29)	72.07(+19.85)	75.29(+14.23)
TestB	85.08(+6.74)	79.93(+7.84)	82.43(+6.93)

Table 14: The promotion to the baselines of sentence punctuation

5.4 Error Analysis

By analyzing the errors made by each team's system, we are able to observe different performances across different genres of texts and different punctuation.

5.4.1 Genres

Table 15 lists the F1 scores of teams in sentence segmentation and punctuation of texts in four genres. It becomes evident that most teams excelled in sentence segmentation and punctuation accuracy with *Products*, followed by county annals, and then academy records, while performance was notably lower with Buddhist sutra. The divergent performance across these four genres are examined as follows.

Firstly, the training set predominantly comprises data from genres such as county annals and academy records, with minimal representation from *Buddhist sutra*. Consequently, teams achieved markedly higher scores in county annals and academy records compared to *Buddhist sutra*, owing to the disparity in data within the training set.

Secondly, most teams gain the highest scores on *Products* data, despite its limited occurence in the training set. This is caused by the prevalence of slight-pauses and commas in *Products* data, typically occurring within lists of words devoid of complex vocabulary or syntactic structures. Example 2 is an example of *Products* data with many slight-pauses. Consequently, the model could achieve superior results on *Products* data through straightforward judgments.

(2) 打鐵鳥、黎母雀、紅頭鶯、鵓鴿、喜鵲、麻雀、山呼、鸚鵡、鴝鵒、秦吉了、五色雀、雉雞、烏、黃鶯、剪刀雀、鷓鴣、鳩、百舌、鵪鶉、杜鵑、畫眉、啄木、火雞、山雞、鴟鴞、蓑衣鶴、水鴨、白臉雞、鷺鷥、青莊、鶻鵃、翡翠、鵜鶘、鸂鶒、鴛鴦、割雀、鷗、海鵝、水鷹、海鳥、鶴、火烏、烏須、天鵝、知風、水晶、飛魚鳥、檳榔燕、華雞。

Team	Products		County		Buddhist		Academy	
	F_{seg}	F_{punc}	F_{seg}	F_{punc}	F_{seg}	F_{punc}	F_{seg}	F_{punc}
BNU	80.36	64.66	85.47	67.78	61.42	47.34	84.47	71.91
CT	**93.58**	**82.20**	**89.46**	73.91	83.44	50.47	87.96	75.6
MiDU	91.66	81.63	88.28	73.21	**85.43**	**71.80**	88.87	**77.43**
NJU1	88.23	73.04	83.23	64.95	74.04	58.95	83.67	70.97
NJU2	75.96	63.73	87.17	**74.10**	76.78	62.53	86.07	75.25
SU	91.38	81.85	88.13	72.13	79.01	61.91	**89.09**	75.46

Table 15: F1 scores for sentence segmentation and punctuation of texts in four genres (%)

5.4.2 Punctuation of Different Types

Table 16 lists the quantity of annotations and corresponding scores for different punctuation marks in the highest-scoring *TestA* submissions by MiDU. In Table 16, *TestA* (gold) means the number of gold punctuation in *TestA*. Machine (Total) means the total number of punctuation tagged by the MIDU's system running on *TestA*. Machine (Correct) means the number of correct punctuation tagged by MIDU's system. It is evident that comma exhibits the highest performance, while double quotation marks and book title punctuation perform less satisfactorily. There are three main issues with the system's performance in punctuation.

Puncs	P (%)	R (%)	F (%)	TestA (gold)	Machine (Correct)	Machine (Total)
、	92.34	71.24	80.43	1,269	904	979
，	77.34	79.23	78.27	4,949	3,921	5,070
。	76.38	76.67	76.52	2,332	1,788	2,341
？	77.5	70.45	73.81	88	62	80
！	93.33	48.28	63.64	29	14	15
；	76.92	45.98	57.55	87	40	52
：	77.12	44.36	56.32	266	118	153
《	87.72	27.78	42.19	180	50	57
》	82.46	26.11	39.66	180	47	57
“	66.67	10.07	17.5	139	14	21
”	63.16	8.82	15.48	136	12	19

Table 16: Punctuation scores by MIDU

First, the number of samples in the training set affects the effectiveness of punctuation annotation. Table 17 shows the distribution of punctuation marks in the training set. In conjunction with Table 16, it can be observed that punctuation marks with better annotation performance, such as commas, are more numerous in the training set, whereas punctuation marks with poorer performance ,such as book title marks, are less frequent. Therefore, to further improve the model's performance, it would be advisable to select different corpora when creating the training set, to adjust the distribution consistency of punctuation marks within the training set.

Puncs	Count
，	1,879,220
。	954,948
：	163,968
、	126,394
“	120,769
”	119,407
？	73,067
《	60,302
》	60,256
；	55,256
！	45,623

Table 17: The distribution of punctuation marks in the training set

Second, the genres also affects the effectiveness of punctuation annotation. Despite the relatively sparse presence of commas in the training set, their strong performance can be attributed to the abundance of commas and periods in the text of *Products* (物产), which makes the annotation poccess easier and more accurate.

Thirdly, the issue of pairing exists in the use of paired punctuation marks. Among the eleven types of punctuation marks, double quotes and book title marks are different from others in that they appear in pairs. These paired punctuation marks have some specific requirements in annotation : after a left quote, there must be a right quote, and not another left quote, and the number of left and right quotes must be the same. However, according to the data in Table 16, the number of left quotes annotated by machine does not equal the number of right quotes. Although 21 left quotes were annotated, only 19 right quotes appeared. This indicates that post-processing can be used to further improve the performance of the systems.

6. Discussion

6.1 Consistency in Paired Punctuation Marks

In the evaluation of various punctuation types within the submissions, a notable inconsistency was observed in the usage of inherently paired punctuation marks, such as double quote marks and book title marks. This inconsistency was particularly evident in one team's submission, where a significant imbalance was recorded: the frequency of left quote marks was nearly fivefold that of right quote marks. Although numerous teams have employed specialized post-processing techniques to address character omission and addition issues common in large language models, these efforts appear to have insufficiently accounted for the nuances of Chinese punctuation. Moreover, a critical oversight in these submissions is the lack of consistency checks for paired punctuation marks. Such checks are essential for ensuring punctuation accuracy, especially in the context of complex language structures like those found in Chinese.

6.2 Implementation Strategy for Book Title Marks

The low performance in handling book title marks, as observed in this evaluation, stems from two main issues: inconsistent handling across different cases, and the approach adopted for processing quote marks. Book title marks, which are used to denote book titles, chapter names, and similar entities, warrant a specific treatment due to their distinct significance. In fact, the annotation of these marks could be effectively treated as a task of named entity recognition, primarily focusing on book titles. Previous studies have approached book title marks as individual named entities, yielding some successful outcomes. However, during this evaluation, it became evident that participating teams did not develop specialized solutions for book title marks. Instead, they handled them as generic punctuation marks and failed to observe their specific function and importance.

6.3 Character Discrepancies Due to Large Language Models

Large language models, particularly generative ones, often alter the original text during prompt engineering, automatically adding or removing Chinese characters, leading to discrepancies between the output and the original text. In this evaluation, most teams encountered issues with character omission and redundancy. The majority of differences of Chinese characters between the submitted results and the test set are around 1% to 2%, with the largest deviation reaching 8%. Although algorithms were employed in this evaluation to rectify the problems of character omission and redundancy in the submissions, teams still struggled to achieve high scores. Hence, to solve the issues of character omission and addition over-generated by large language models, post-processing is needed for the text consistancy. Another way is to constrain the generated characters during model output generation to maintain consistency with the original text.

7. Conclusions

EvaHan2024 marks a pioneering endeavor in the field of ancient Chinese sentence segmentation and punctuation. The best system of this bakeoff, developed by MiDU, notably outperformed the majority of its counterparts. The deployment of large language models has indeed elevated performance metrics in processing ancient Chinese texts. The test sets have a wide coverage and one was implemented as a blind test, therefore, the effectiveness of sentence segmentation and punctuation is more challenging than expected, leaving ample room for improvement. It is noteworthy that even advanced language models are not immune to issues such as character omission and excessive generation. Therefore, it is imperative for participating teams to actively engage with and address these complexities. In the future, the next iteration of EvaHan should broaden its scope to encompass a wider array of genres and cross-temporal corpora. This expansion is anticipated to foster improvements in handling a more diverse set of data.

8. Acknowledgements

Thank the reviewers for their advices. Thank Yongji Wang, Jingxuan Xi, Pengxiu Lu, Ruijia Yang, and Han Xiao for their data annotation and checking. Thank GULIAN (Beijing) Media TECH CO.,LTD for their data support. This research was supported by National Social Science Funds of China (21&ZD331), and National Language Commission Project of China (YB145-41).

9. References

Bakare, A. M., Anbananthen, K. S. M., Muthaiyah, S., Krishnan, J., and Kannan, S. (2023). Punctuation Restoration with Transformer Model on Social Media Data. *Applied Sciences*, 13(3), 1685.

Chen, T., Chen, R., Pan, L., Li, H., and Yu, Z. (2007). Archaic Chinese Punctuating Sentences Based on Context N-gram Model. *Computer Engineering* (03), 192-193+196.

Cheng, N., Li, B., Ge, S., Hao, X., and Feng, M. (2020). A Joint Model of Automatic Sentence Segmentation and Lexical Analysis for Ancient ChineseBased on BiLSTM-CRF Model. *Journal of Chinese Information Processing*, 34(04), 1-9.

General Administration of Quality Supervision, Inspection and Quarantine of the People's Republic of China. (2012). *General Rules for Punctuation* (GB/T 15834-2011). Beijing: Standards Press of China.

Hong, T., Cheng, R., Liu, S., and Fang, K. (2021). An Automatic Punctuation Method Based On the Transformer Model. *Digital Humanities* (02), 111-122.

Hou, H. and Huang, J. (2008). On Sentence Segmentation and Punctuation Model for Ancient Books on Agriculture. *Journal of Chinese Information Processing* (04), 31-38.

Hu, R., Li, S., and Zhu, Y. (2021). Knowledge Representation and Sentence Segmentation of Ancient Chinese Based on DeepLanguage Models. *Journal of Chinese Information Processing*, 35(04), 8-15.

Jin, G. and Chen., X. (2008). The Fourth International Chinese Language Processing Bakeoff: Chinese Word Segmentation, Named Entity Recognition and Chinese POS Tagging. *In Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*.

Jones, D. A., Wolf, F., Gibson, E., Williams, E., Fedorenko, E., Reynolds, D. A., and Zissman, M. (2003). Measuring the Readability of Automatic Speech-To-Text Transcripts. *8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, 1585–1588.

Kolár, J., and Lamel, L. (2012). Development and evaluation of automatic punctuation for french and english speech-to-text. *Interspeech*, 1376-1379.

Li, B., Yuan, Y., Lu, J., Feng, M., Xu, C., Qu, W., and Wang, D. (2022). The First International Ancient Chinese Word Segmentation and POS Tagging Bakeoff: Overview of the Evahan 2022 Evaluation Campaign. *Proceedings of the second workshop on language technologies for historical and ancient languages*, 135-140.

Lyu S. (1983). The First Step in Organizing Ancient Texts. *China Publishing Journal*, 71(4), 44-50.

Meister, A., Novikov, M., Karpov, N., Bakhturina, E., Lavrukhin, V., and Ginsburg, B. (2023). *LibriSpeech-PC: Benchmark for Evaluation of Punctuation and Capitalization Capabilities of end-to-end ASR Models*. *Proceddings of 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*: 1-7.

Nagy, A., Bial, B., and Ács, J. (2021). Automatic punctuation restoration with BERT models. arXiv:2101.07343v1.

Peitz, S., Freitag, M., Mauser, A., and Ney, H. (2011). Modeling Punctuation Prediction as Machine Translation. *Proceddings of the International Workshop on Spoken Language Translation*, 238–245.

Shi, X., Shi, X., Shi, X., Shi, X., and Song, Y. (2019). A Method and Implementation of Automatic Punctuation. *Journal of Digital Archives and Digital Humanities* (3), 1-19.

Sommerschield, T., Assael, Y., Pavlopoulos, J., Stefanak, V., Senior, A., Dyer, C., Bodel, J., Prag, J., Androutsopoulos, I., and Freitas, N. D. (2023). Machine Learning for Ancient Languages: A Survey. *Computational Linguistics*, 1-45.

Su, Q., Hu, R., Zhu, Y., Yan, C., and Wang, J. (2021). Key Technologies for Digitization of Ancient Chinese Books. *Digital Humanities Research* (03), 83-88.

The State Administration of Press, Publication, Radio, Film and Television of the People's Republic of China. (2015). *Academic Publishing Specification-Collation of Chinese Ancient Books* (CY/T 124-2015). Beijing: Standards Press of China.

Tündik, M. Á., Szaszák, G., Gosztolya, G., and Beke, A. (2018). User-centric Evaluation of Automatic Punctuation in ASR Closed Captioning. *Interspeech* 2018, 2628–2632.

Wang, D., Lin, L., Zhao, Z., Ye, W., Meng, K., Sun, W., Zhao, L., Zhao, X., Shen, S., Zhang, W., and Li, B. (2023). EvaHan2023: Overview of the First International Ancient Chinese Translation Bakeoff. *Proceedings of ALT2023: Ancient Language Translation Workshop*, 1-14.

Yu, J., Wei, Y., and Zhang, Y. (2019). Automatic Ancient Chinese Texts Segmentation Based on BERT. *Journal of Chinese Information Processing* (11), 57-63.

Yuan, Y., Li, B., Feng, M., He, S., and Wang, D. (2022). A Joint Model of Automatic Sentence Segmentation and Punctuation for Ancient Classical TextsBased on Deep Learning. *Library and Information Service*, 66(22), 134-141.

Two Sequence Labeling Approaches to Sentence Segmentation and Punctuation Prediction for Classic Chinese Texts

Xuebin Wang and Zhenghua Li
School of Computer Science and Technology, Soochow University, China
xbwang15@stu.suda.edu.cn; zhli13@suda.edu.cn

Abstract

This paper describes our system for the EvaHan2024 shared task. We design and experiment with two sequence labeling approaches, i.e., one-stage and two-stage approaches. The one-stage approach directly predicts a label for each character, and the label may contain multiple punctuation marks. The two-stage approach divides punctuation marks into two classes, i.e., pause and non-pause, and separately handles them via two sequence labeling processes. The labels contain at most one punctuation marks. We use pre-trained SikuRoBERTa as a key component of the encoder and employ a conditional random field (CRF) layer on the top. According to the evaluation metrics adopted by the organizers, the two-stage approach is superior to the one-stage approach, and our system achieves the second place among all participant systems.

Keywords: EvaHan2024, Sentence Segmentation, Punctuation Prediction, Sequence Labeling

1. Introduction

One important characteristic of classic Chinese texts if the lack of punctuation marks. Readers have to decide sentence boundaries. In consequence, an article in classic Chinese is usually much more ambiguous than that in modern Chinese. The goal of the EvalHan2024 shared task is to see whether computation models can automatically perform sentence segmentation (SS) and punctuation prediction (PP).

We design and experiment with two sequence labeling approaches, i.e., one-stage and two-stage approaches. The one-stage approach is quite straightforward. It directly predicts a label for each character, and the label may contain multiple punctuation marks, as shown in the bottom row in Figure 2.

For the two-stage approach, we distinguish two types of punctuation marks, i.e., pause and non-pause, as shown in Table 1 . Then, we predict the two types of punctuation marks using two separate sequence labeling models. For both models, each label contains at most one punctuation mark.

Pause marks corresponds to those indicating sentence boundaries. Therefore, once the punctuation marks are obtained, we can infer sentence boundaries. Therefore, we only focus on the PP subtask, and solve the SS subtask as byproduct.

For the model architecture, we employ a standard conditional random field (CRF) model, using SikuRoBERTa as a key component of the encoder, as shown in Figure 1.

According to the evaluation metrics adopted by the organizers, the two-stage approach is superior to the one-stage approach, and our system achieves the second place among all participant systems. Compared to the baseline model Xunzi-Qianwen-7B-CHAT, our models obtain large improvement. Our code is available at `https://github.com/XuebinWang-ai/EvaHan2024_PP`.

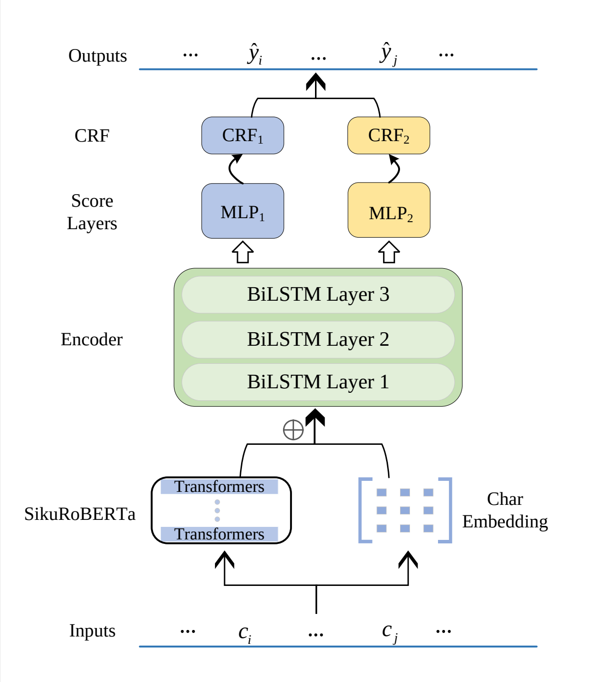
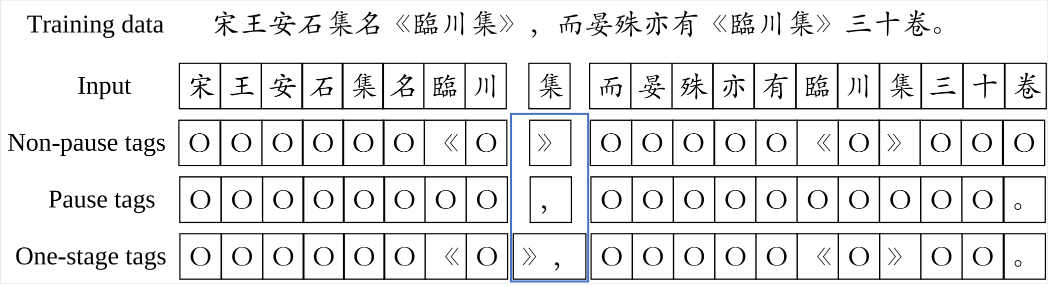


Figure 1: Model architecture.

2. Related Works

Sentence Segmentation & Punctuation Prediction A work by Xu et al. (2019) demonstrates combining word embedding and radical embedding can enhance the LSTM-CRF model in the SS task. A research by Hu et al. (2021) indicates a notable improvement in the performance of the BERT language model (Devlin et al., 2019) compared to the BiLSTM-CRF model in the SS task,



Training data 宋王安石集名《臨川集》，而晏殊亦有《臨川集》三十卷。

Input	宋	王	安	石	集	名	臨	川	集	而	晏	殊	亦	有	臨	川	集	三	十	卷
Non-pause tags	O	O	O	O	O	O	《	O	》	O	O	O	O	O	《	O	》	O	O	O
Pause tags	O	O	O	O	O	O	O	O	，	O	O	O	O	O	O	O	O	O	O	。
One-stage tags	O	O	O	O	O	O	《	O	》，	O	O	O	O	O	《	O	》	O	O	。

Figure 2: This excerpt is from the pre-processed training dataset. Punctuation marks are typically annotated on the characters to their left, apart from three specific types of left punctuation marks, which are annotated on the characters to their right. The “O” tags represent positions without punctuation.

Pause marks		Non-pause marks	
Name	Punc	Name	Punc
Comma	，	Double Quotation	“ ”
Period	。	Single Quotation	‘ ’
Slight-pasue	、	Book Title Marks	《 》
Question	？		
Exclamation	！		
Colon	：		
Semiclon	；		

Table 1: Pause and non-pause punctuation marks.

Symmetrical pairs			
Punc pair	Number	Punc pair	Number
。”	55580	”。	3293
？”	17878	”？	63
！”	8447	”！	32
。’	1945	’。	417
。》	843	》。	3043
，”	138	”，	6899
，》	35	》，	4957

Table 2: High frequency punctuation pairs.

resulting in a remarkable 10% increase in the F1 score. Conversely, a study by YuJ highlightes that the use of the BERT-BiLSTM-CRF model slightly improves the PP task performance over the BERT-CRF model. However, post-incremental training with an extensive corpus of traditional Chinese texts improves the performance of BERT for these two tasks, in relation to the BERT-CNN and BERT-CRF models (Tang et al., 2021).

Pre-trained Model The BERT model has gained significant prominence in various Chinese language processing tasks, including word segmentation, part-of-speech tagging, among others. Nonetheless, it is essential to note that BERT’s pre-training primarily focuses on Simplified Chinese while SikuRoBERTa (Wang et al., 2022) on traditional Chinese texts. Consequently, SikuRoBERTa performs better in the situation of dealing with classical Chinese texts.

3. Our Method

In this section, we introduce our methods and model architectures.

The EvaHan2024 task encompasses two subtasks, i.e., the SS subtask and the PP subtask. Sentence boundaries are closely correlated with some punctuation marks, such as periods and exclamation marks. We call these punctuation marks *pause marks*. We call other punctuation marks *non-pause marks*. Table 1 lists the two types of punctuation marks.

Upon distinguishing the two types of punctuation marks, we propose to avoid the SS subtask and treat it as a part of the PP subtask. Moreover, we handle the two types of punctuation marks separately via sequence labeling.

3.1. Data Pre-processing

Figure 2 illustrates how to pre-process raw training data. The character sequence without punctuation marks composes an input sequence for the two independent sequence labeling models. The middle two rows give the tag sequences for the two models.

3.2. Two stages

The above pre-processing method leads to the problem of being unable to determine the order during post-processing when two CRFs predict marks at the same position. The high-frequency punctuation pairs in Table 2 illustrate that this problem cannot be avoided. We propose two methods to solve this problem.

Two-stage Method When we divide the punctuation points into two groups, we improve on the post-processing method. We counts the frequency of different orders from the training set, and selects the order with higher frequency as the final result[1].

One-stage Method The one-stage method is to dropout the label grouping method and treat the PP task as one sequence labeling task instead of two. Specifically, we treat punctuation combinations that appear at the same position as one label. Moreover, some low-frequency labels can be mapped to high-frequency labels to simplify the label set.

We compare the performance of these two approaches in Table 5.

3.3. Models

The input sequence is defined as $S = \{c_0, c_1, ..., c_n\}$, where n represents sequence length and c_i denotes the i-th character of the sequence. The lowest embedding layer of the model utilizes SikuRoBERTa and character embedding.

The SikuRoBERTa output representation of character c_i is denoted as $\boldsymbol{e}_i^s$. The character embedding representation of character c_i is denoted as $\boldsymbol{e}_i^c$. The concatenation of $\boldsymbol{e}_i^s$ and $\boldsymbol{e}_i^c$ forms the embedding representation of character c_i, expressed as $\boldsymbol{e}_i$. The formulation of this representation is as follows:

$$\boldsymbol{e}_i = \boldsymbol{e}_i^s \oplus \boldsymbol{e}_i^c \tag{1}$$

After obtaining the embedding layer representation, it is encoded through three BiLSTM layers to derive the contextual representation.

$$\boldsymbol{R} = BiLSTMs(\boldsymbol{e}) \tag{2}$$

Within this framework, $\boldsymbol{e}$ signifies the embedding representation of the input sequence, while $\boldsymbol{R}$ is the context representation.

The final two layers consist of distinct MLP-CRF models. The MLP layer extracts information from the contextual representation and reduces the vector dimension to match the size of the label set.

$$\boldsymbol{S} = MLP(\boldsymbol{R}) \tag{3}$$

In this formula, $\boldsymbol{S}$ denotes the outputs of the MLP model.

Subsequently, the CRF layer calculates the CRF-loss during training and employs the Viterbi algorithm for inference purposes. The implementation of the CRF model is based on SuPar[2].

Data parameters	Numbers
Train set lines	263,091
Dev set lines	13,984
Chars	10,638
Max length	510
Window size	100
Tag combinations	160
Tag combinations of one-stage	72
Non-Pause tags	40
Pause tags	7

Table 3: Parameters after data processing.

Hyperparameters	Values
Dimension of SikuRoBERTa	100
Dimension of char embedding	100
Hidden dimension of BiLSTM	400
Dimension of MLP1	41
Dimension of MLP2	8
Learning rate of BiLSTM	2e-5
Learning rate of MLPs and CRFs	2e-4
Dropout ratios	0.33
Batch size	50

Table 4: Hyperparameters.

$$loss = crf\_loss(\boldsymbol{S}, y) \tag{4}$$

$$\hat{y} = Viterbi(\boldsymbol{S}) \tag{5}$$

In this context, y represents the ground truth while $\hat{y}$ signifies the prediction result.

4. Experiments

4.1. Data

In this task, the training data shared by EvaHan2024 originates from the *Siku Quanshu*, containing over 10 million characters. We designate 5% of the training data as provisional validation data for assessing the model's performance. Furthermore, in addition to this dataset, we employe the Xunzi-Qianwen-7B-CHAT to generate approximately 11,000 synthetic classical Chinese sentences. These generated data are utilized for both training and validation purposes.

The handling of long sequences poses a challenge. As these sequences represent a minority in the training data, they are typically truncated directly. For evaluation, we employs the parallel sliding window approach described in Tang et al. (2021) to manage using a fixed window size, without compromising efficiency and performance.

The parameters of processed dataset is shown in Table 3. The "Tag combinations" entry in Table 3 comprises a count of 160. This figure is the

[1] In fact, we did not use this method when submitting the results, but rigidly placed all pause marks after non-pause marks. While this does not affect the calculated F1 score, we have modified this in the published code.

[2] Supar Github: https://github.com/yzhangcs/parser.

Test A	Sentence Segmentation			Punctuation Prediction		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Baseline	90.53	66.12	76.42	73.52	52.22	61.06
ChatGPT-3.5	83.81	59.85	69.83	63.90	43.88	52.03
Our Model (One-stage)	91.23	83.25	87.06	76.41	67.88	71.89
Our Model (Two-stage)	89.82	84.69	**87.18**	75.87	69.70	**72.66**
Test B	Sentence Segmentation			Punctuation Prediction		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Baseline	95.28	87.17	91.04	79.25	72.09	75.50
Our Model (One-stage)	95.47	91.47	**93.43**	83.42	78.42	80.84
Our Model (Two-stage)	94.64	91.93	93.27	82.93	78.96	**80.89**

Table 5: Test set results. Test B is implemented on the Zuozhuan test set.

total number of punctuation combinations present in the dataset when they are not separately labeled. Upon labeling according to the classification method mentioned in Section 3, the size of the label set can be notably diminished to 40 and 7.

4.2. Results

The training hyperparameters are detailed in Table 4, with the Adam optimizer employed. The model training is conducted on an NVIDIA Tesla-V100-SXM2-32G GPU, utilizing a batch size of 50 which requires approximately 30G of memory per iteration. Each iteration takes 4.5 hours. Notably, it is observed that the model achieves optimal performance on the validation set in the 4th iteration.

In accordance with common practice, the evaluation of our model entails assessing its Precision (P), Recall (R), and F1 score. The results are presented in Table 5, it can be seen that the two-stage method performs better on the test set. The experimental results demonstrate that the task performance of our model vastly outperform the baseline model on both evaluation sets.

5. Discussion

In this task, our model shows robust performance, owing to several enhancements.

Firstly, we distinguish between non-pause and pause punctuation to simplify the process of sequence labeling. Secondly, introducing SikuRoBERTa and character embeddings into the model architecture to obtain embedding representations. In addition, we employ XunziALLM to generated classical Chinese writings for training and validation.

However, there are flaws in our approach.

Firstly, the two-stage method we mentioned in Section 3 is not elegant. Another idea is to train a binary classifier to determine the order. Secondly, an issue of incomplete data processing arises due to the expansive nature of the dataset and encoding difficulties associated with some traditional Chinese characters. Consequently, instances of missing characters or incomplete sentence are encountered. We treat these data as noise and remove them. Furthermore, we apply the rule-based method to correct the illegal punctuation marks within the dataset. It is acknowledged, however, that the efficacy of this correction method is limited. Thirdly, The BiLSTM layers process lengthy texts slowly, lengthen the training process. Moreover, The XunziALLM tool is not fully leveraged.

Acknowledgements

We thank organizers of the EvaHan2024 shared task for their help and hard work, all the anonymous reviewers for their valuable comments, and Jielin Chen for her help in improving the writing of this paper. This work was supported by National Natural Science Foundation of China (Grant No. 62176173 and 62336006), and a Project Funded by the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions.

6. References

Ning Cheng, Bin Li, Liming Xiao, Changwei Xu, Sijia Ge, Xingyue Hao, and Minxuan Feng. 2020. Integration of automatic sentence segmentation and lexical analysis of Ancient Chinese based on BiLSTM-CRF model. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 52–58, Marseille, France. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Renfen Hu, Shen Li, and Yuchen Zhu. 2021. Knowledge representation and sentence segmentation of ancient chinese based on deep language model. *Journal of Chinese Information Processing*, 35(4):8–15.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Icml*, volume 1, page 3. Williamstown, MA.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xuemei Tang, Qi Su, Jun Wang, Yuhang Chen, and Hao Yang. 2021. Automatic traditional Ancient Chinese texts segmentation and punctuation based on pre-training language model. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 678–688, Huhhot, China. Chinese Information Processing Society of China.

Dongbo Wang, Chang Liu, Zihe Zhu, Jiangfeng Liu, Haotian Hu, Si Shen, and Bin Li. 2022. Construction and application of pre-trained models of siku quanshu in orientation to digital humanities. *Library Tribune*, 42(6):31–43.

Han Xu, Wang Hongsu, Zhang Sanqian, Fu Qunchao, and Liu Jun. 2019. Sentence segmentation for classical chinese based on lstm with radical embedding. *The Journal of China Universities of Posts and Telecommunications*, 26(2):1.

Ancient Chinese Sentence Segmentation and Punctuation on Xunzi LLM

Shitu Huo, Wenhui Chen
Beijing Normal University, Xiamen University
Beijing, Fujian, China
saitofok@gmail.com, 793994571@qq.com

Abstract

This paper describes the system submitted for the EvaHan2024 Task on ancient Chinese sentence segmentation and punctuation. Our study utillizes the Xunzi large language model as the base model to evaluate the overall performance and the performance by record type. The applied methodologies and the prompts utilized in our study have shown to be helpful and effective in aiding the model's performance evaluation.

Keywords: Natural Language Processing, Ancient Chinese, Sentence Segmentation, Punctuation

1. Introduction

Throughout history, Chinese civilization has given birth to countless invaluable classics, imbued with rich philosophical thought, historical records, and literary enlightenment. These ancient texts are not only the crystallization of the Chinese nation's precious wisdom but also an integral part of the common heritage of human civilization. However, due to the significant differences between ancient Chinese and modern Chinese in terms of grammar, vocabulary, and semantics, the digitalization and automatic computational understanding of these ancient texts pose tremendous challenges.

In the process of digitizing ancient texts, accurate sentence segmentation and punctuation are crucial steps. Reasonable sentence segmentation can enhance the reading experience and lay the groundwork for subsequent semantic analysis. However, because ancient texts contain a large number of unique grammatical constructions and rhetorical devices, traditional sentence segmentation and punctuation often rely on manual processing by experts, which is time-consuming and laborious. Therefore, developing automated evaluation models and algorithms is an urgent need to improve efficiency and quality.

EvaHan2024 is an international evaluation currently focusing on automatic sentence parsing and punctuation assessment tasks in Classical Chinese. This research proposes to utilize the Xunzi large language model and tailor prompt engineering strategies specifically on sentence segmentation and punctuation for ancient Chinese. As a result, we have a relatively higher performance than baseline with effective prompts.

2. Related Study

2.1 Study on Statistical Machine Learning and Deep Learning Methods for Segmentation and Punctuation for Ancient Chinese

Segmentation mainly divides into rule-based methods and statistical methods. Rule-based methods are typically formulated by experts in ancient Chinese, using common linguistic knowledge to help construct a system for sentence segmentation. For example, segmentation can be based on antonymous compound words, book citation markers, numerals, reduplicated words, and verb-noun structures (Huang & Hou, 2008). However, actual sentence segmentation is very complex, with a word having multiple meanings and combinations, making it impossible to segment based solely on a single word or combination. Rule-based methods cannot cover all situations, leading to scenarios akin to Gödel's incompleteness theorems.

Statistical methods were subsequently widely used. Early experiments could use n-grams (Chen et al., 2007), Conditional Random Fields (Zhang et al., 2009), and the relationship features between adjacency collocation intensities (Xu, 2011) for judgment. Later, scholars increasingly turned to deep learning methods, with BERT being one of the most widely utilized models. BERT (Bidirectional Encoder Representation Transformers) shows excellent performance at language inference and other NLP tasks (Devin et al., 2018). Yu et al. (2019) used BERT for ancient Chinese sentence segmentation research, achieving better results than the BiLSTM+CRF model. Wei (2020) fine-tuned the BERT model, achieving F1 scores of 70.40% for punctuation and 91.67% for segmentation on a large-scale composite corpus. Hu et al. (2021) compared the sequence labeling methods of BERT+FCL, BERT+CRF, and BERT+CNN on the task of ancient Chinese sentence segmentation, finding that BERT+CNN had the best automatic sentence segmentation performance in the three literary forms of poetry, ci, and ancient prose, reaching F1 scores of 99%, 95%, and 92%, respectively. Tang et al. (2023) used a large-scale traditional ancient Chinese corpus to incrementally train the BERT Chinese model, achieving automatic sentence segmentation F1 scores of 95.03% and 99.53% for ancient prose and poetry, respectively, and automatic punctuation F1 scores of 80.18% and 98.91%, respectively.

In conclusion, the evolution of methodologies in ancient Chinese sentence segmentation has shown a clear trajectory from rule-based approaches towards the adoption of deep learning techniques.

2.2 Study on Large Language Models for Ancient Chinese

With the successful implementation of scaling laws on large language models, these models have been able to grasp the deep semantics and grammatical rules of languages. Several studies have recently focused on evaluating the capabilities of large language models (LLMs) in comprehending ancient languages, with a particular emphasis on ancient Chinese. One notable contribution in this area is the work by Zhang and Li (2023), who introduced ACLUE, an evaluation benchmark designed specifically to assess LLMs' language abilities in relation to ancient Chinese. ACLUE comprises 15 tasks covering various linguistic skills, including phonetic, lexical, syntactic, semantic, inference, and knowledge. Notably, ChatGLM2 exhibited the highest performance level among the evaluated models, achieving an average accuracy of 37.45%.

Currently, the existing ancient Chinese large models include AI Jiusi, AI Taiyan, and the Xunzi model, which are mainly based on existing pre-trained models and fine-tuned on ancient Chinese datasets.

AI Jiusi is a large model fine-tuned by Huazhong University of Science and Technology based on the Alibaba Cloud Tongyi Qianwen as the base model[1].

AI Taiyan is a large language model specifically designed for understanding Classical Chinese texts, developed by the Digital Humanities Department at Beijing Normal University[2].

The Xunzi large language model includes versions fine-tuned on Qwen-7B, GLM-6B, Baichuan-7B for Classical Chinese[3]. In summary, the above models have not yet undergone comprehensive evaluation on segmentation and punctuation benchmarks, necessitating further exploration.

3. Employed Model

Qwen-7B is a large language model based on the Transformer architecture, trained on an extensive pre-training dataset (Bai et al., 2023). Xunzi large model is fully fine-tuned based on Qwen-7B. The training of this model utilized the Zero2 technology in the DeepSpeed framework for memory optimization, distributing the model's state parameters and gradients across 8 A800 model GPUs. The fine-tuning dataset comprised approximately 5GB of ancient text corpora mixed with modern Chinese texts, command data, and other types of corpora, thus creating a mixed dataset containing 4 billion Chinese characters.

This study employs the Xunzi large model and deploy it to a server and conduct large-scale evaluations on various text datasets to assess its practical utility and scalability.

4. Experiment

4.1 Experimental Environment

The NVIDIA card is configured in Table 1:

CUDA Version	GPU	Memory
12.1	NVIDIA GeForce RTX 4090	24GB

Table 1: The Nvidia Info

4.2 Prompt Engineering

This study explores how carefully designed prompts can guide Xunzi model to generate more accurate and rich text content. The method employed in this study adopts a two-round prompt design strategy, aimed at refining and optimizing the final output through the text generated initially.

In the first round, we designed an initial prompt: "Please think boldly, and as diversely and richly as possible, punctuate the following text, and respond in traditional characters: {text}." The goal of this prompt is to guide the model to process a complex sentence, making the meaning of the text clearer by adding appropriate punctuation, while maintaining the diversity and richness of sentence structure. In this stage, the model was run five times, generating five different punctuated results.

Following this, in the second round, another prompt was adopted: "Please consider and integrate the optimal sentence breaking scheme from the following five sentences: {response}." This step requires the model to select and integrate the best sentence breaking scheme from the five punctuated sentences generated in the first round[4]. In the end, we collect all the best sentences from the test set.

1 https://mp.weixin.qq.com/s/c-NeKg4z4dMgBSFUbYDtbg

2 https://mp.weixin.qq.com/s/Cp5NOSOcjvBt9qzcVZ9igQ

3 https://github.com/Xunzi-LLM-of-Chinese-classics/XunziALLM

4 The actual prompt is in Chinese. The first prompt is: "请大胆思考，尽可能多样、丰富地给下面的文本打上标点符号, 请使用繁体字回答：{text}." The second prompt is : "请从下列五个句子中，思考并整合出最优的断句结果：{response}."

4.3 Temperature Setting

We systematically varied the parameter known as 'temperature', a scaling factor applied to the logits of the model before sampling. The temperature setting is shown in Table 2. During the initial round of interaction, the temperature was set to 0.95, promoting a diverse and creative output by allowing for a broader probability distribution of potential responses. Subsequently, in the second round, the temperature was reduced substantially to 0.1, significantly narrowing the scope of variability in the model's output. This reduction in temperature typically yields more deterministic and possibly repetitive results, given the higher likelihood of sampling the most probable outcomes. This methodological adjustment of the temperature parameter is critical in fine-tuning the model's performance to align with the desired level of creativity and variability in the generated content.

Temperature	Value
first round	0.95
second round	0.1

Table 2: The Parameter of the Model

5. Results

5.1 Overall Performance

Test sets include Test A and Test B. Test A refers to the data released the first time while Test B refers to the Zuozhuan data released the second time.

The performance metrics in Test A presented in Table 3 underscore the relative strengths and weaknesses of different models in handling segmentation and punctuation tasks for text analysis. The segmentation task, as evident from the data, benefits from higher accuracy across all models when compared to punctuation.

Our prompt engineering method based on Xunzi-Qwen-7B model outstrips its predecessors, achieving a precision of 90.70% in segmentation, indicating exceptional reliability in predicting segment boundaries. However, a recall of 71.54% suggests room for improvement in identifying all true segment boundaries. The F1-score, at 79.99%, represents a favorable balance between precision and recall, underscoring a robust segmentation model. In comparison, GPT-3.5 and Xunzi-Qwen-7B demonstrate precision rates of 83.81% and 90.53%, respectively, with the latter nearly matching our model. However, both models fall short in recall, and consequently, F1-scores, with GPT-3.5 at 59.85% and 69.83%, and Xunzi-Qwen-7B at 66.12% and 76.42%, respectively.

For the punctuation task, the results indicate more challenges across the board. Our method achieves a precision of 73.63%, suggesting a correct prediction in approximately three out of four instances. Yet, the recall of 56.86% reveals that the model fails to detect a significant number of true punctuation marks. This is reflected in the F1-score, which at 64.17%, points to moderate overall effectiveness. GPT-3.5's punctuation capability is weaker still, with precision and recall scores of 63.90% and 43.88%, respectively, and an F1-score of 52.03%. Xunzi-Qwen-7B presents comparable results to our model in precision at 73.52% but lags in recall at 52.22%, culminating in an F1-score of 61.06%.

Method	Task	Precision	Recall	F1-score
Ours	Seg	**90.70%**	**71.54%**	**79.99%**
GPT-3.5		83.81%	59.85%	69.83%
Baseline (Xunzi-Qianwen-7B-CHAT)		90.53%	66.12%	76.42%
Ours	Punc	**73.63%**	**56.86%**	**64.17%**
GPT-3.5		63.90%	43.88%	52.03%
Baseline (Xunzi-Qianwen-7B-CHAT)		73.52%	52.22%	61.06%

Table 3: Experiment Results on Test A of Ours(our prompt engineering methods on Xunzi-Qwen-7B), GPT-3.5, Baseline (Xunzi-Qianwen-7B-CHAT).

The performance metrics in Test B, as shown in Table 4, highlight our method's competitiveness against the baseline. Our segmentation model achieved a precision slightly lower than the baseline but exhibited higher recall, indicating a trade-off between precision and recall. Similarly, for the punctuation task, our model demonstrated a balanced trade-off between precision and recall compared to the baseline, suggesting comparable performance between the two models.

Method	Task	Precision	Recall	F1-score
Ours	Seg	95.25%	**88.15%**	**91.57%**
Baseline (Xunzi-Qianwen-7B-CHAT)		**95.28%**	87.17%	91.04%
Ours	Punc	79.06%	**73.66%**	**76.26%**
Baseline (Xunzi-Qianwen-7B-CHAT)		**79.25%**	72.09%	75.50%

Table 4: Experiment Results on Test B of Ours(our prompt engineering methods on Xunzi-Qwen-7B) and Baseline (Xunzi-Qianwen-7B-CHAT).

5.2 Performance by Specific Record Type

The results further show the model's performance on four different types of records (Table 5): Products in Local Products in Local Chronicles(方志物产), County Annals(县志), Buddhist Sutra(佛经), and Academy Records(书院志).

For Products in Local Chronicles, the model achieved a high segmentation precision of 94.78% and a moderate recall of 69.81%, demonstrating high reliability in detecting segments, yet missing some. The punctuation precision was decent at 78.09%, outperforming the recall at 55.22%.

County Annals saw slightly lower segmentation precision but a higher recall, indicating a more balanced performance, and also led the records with the highest punctuation F1-score.

However, Buddhist Sutra presented considerable challenges, with the lowest performance metrics including a segmentation recall of just 46.72%, suggesting the model frequently missed segment points, and the punctuation F1-score fell to 47.23%.

Lastly, Academy Records achieved relatively high segmentation scores and better punctuation performances, although still not surpassing the punctuation results of County Annals. This analysis indicates that while the model shows competency in segmentation, its performance in punctuation is less consistent and requires targeted improvements, particularly within the more complex texts like Buddhist Scriptures.

		Precision	**Recall**	**F1-score**
Products in Local Chronicles	Seg	94.78%	69.81%	80.4%
	Punc	78.09%	55.22%	64.7%
County Annals	Seg	89.61%	81.32%	**85.26%**
	Punc	72.01%	63.66%	67.58%
Buddhist Sutra	Seg	89.02%	46.72%	61.28%
	Punc	68.92%	35.92%	47.23%
Academy Records	Seg	90.78%	78.78%	84.36%
	Punc	77.24%	67.18%	71.86%

Table 5: Our Method's Performance by Record Type

6. Conclusions

Our method is generally more effective at segmenting than punctuating, indicating the need for further training or a different approach for punctuation.

The notably lower performance on Buddhist Scriptures could be due to various factors such as language complexity, formatting, or the presence of Sanskrit or Pali words. Tailored solutions, like adding more scriptural training data or using a specialized tokenization approach.

Improving punctuation accuracy through contextual understanding integration could significantly enhance the model's performance, particularly in ancient texts. Thorough error analysis can uncover specific challenges, while targeted improvements address discrepancies, enhancing consistency and accuracy.

7. References

Bai, J., Bai, S., Chu, Y., et al. (2023). Qwen technical report. *arXiv preprint arXiv:2309.16609.*

Chen, T. Y., Chen, R., Pan, L. L., et al. (2007). Archaic Chinese punctuating sentences based on context n-gram Model. *Computer Engineering, 33*(03), 192-193.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

Hu, R. F., Li, S., & Zhu, Y. C. (2021). Knowledge representation and sentences segmentation of ancient Chinese based on deep language models. *Journal of Chinese Information Processing*, 35(04), 8-15.

Huang, J. N., & Hou, H. Q. (2008). On sentence segmentation and punctuation model for ancient books on agriculture. *Journal of Chinese Information Processing, 22*(04), 31-38.

Tang, X. M., Su, Q., Wang, J., et al. (2023). Automatic traditional ancient Chinese texts segmentation and punctuation based on pre-trained language model. *Journal of Chinese Information Processing, 37*(08), 159-168.

Wei, Y. (2020). *Research on Automatic Texts Segmentation and Word Segmentation For Ancient Chinese Texts.* Beijing: Master Dissertation, Peking University.

Xu, J. Y. (2011). *Research on Automatic Sentence Reading of Ancient Chinese Texts*. Beijing: Ph.D. Dissertation, Peking University.

Yu, J. S., Wei, Y., & Zhang, Y. W. (2019). Automatic ancient Chinese texts segmentation based on BERT. *Journal of Chinese Information Processing, 33*(11), 57-63.

Zhang, K. X., Xia, Y. Q., & Yu, H. (2009). CRF-based approach to sentence segmentation and punctuation for ancient Chinese prose. *Journal of Tsinghua University (Science and Technology), 49*(10), 1733-1736.

Zhang, Y., & Li, H. (2023). Can Large Language Model Comprehend Ancient Chinese? A Preliminary Test on ACLUE. In *Proceedings of the Ancient Language Processing Workshop* (pp. 80–87). Varna, Bulgaria: INCOMA Ltd., Shoumen, Bulgaria.

Sentence Segmentation and Sentence Punctuation based on XunziALLM

Zihong Chen
Nanjing University, China
chenzihong_gavin@foxmail.com

Abstract

In ancient Chinese books, punctuation marks are typically absent in engraved texts. Sentence segmentation and punctuation heavily rely on the meticulous efforts of experts and scholars. Therefore, the work of automatic punctuation and sentence segmentation plays a very important role in promoting ancient books, as well as the inheritance of Chinese culture. In this paper, we present a method for fine-tuning downstream tasks for large language model using the LoRA approach, leveraging the EvaHan2024 dataset. This method ensures robust output and high accuracy while inheriting the knowledge from the large pre-trained language model Xunzi.

Keywords: sentence segmentation, sentence punctuation, ancient Chinese information processing

1. Introduction

Chinese classical texts hold tremendous value as sources of literature. The compilation and organization of these ancient works not only serve as a bridge between the present and the past but also contribute to the scholarly exploration of cultural heritage. However, ancient Chinese writings generally lacked punctuation marks. Consequently, many surviving classical texts lack proper sentence segmentation and punctuation. This poses a significant challenge for readers seeking to comprehend these texts, as well as for scholars engaged in their analysis and interpretation.

Sentence segmentation refers to the process of converting continuous text into a sequence of sentences, where each sentence is separated by a single space. Furthermore, sentence punctuation involves placing the correct punctuation marks at the end of each sentence. However, in classical Chinese texts, sentence punctuation serves the function of sentence segmentation itself, as punctuation marks inherently possess the ability to separate sentences.

Given this situation, an effective automated algorithm needs to be proposed for batch Chinese text segmentation and punctuation tasks. In this paper, we describe the method we used in EvaHan2024. Our system is based on XunziALLM, which is a large pre-trained language base model for ancient Chinese processing. We executed extra training on the fixed provided dataset from classical sources, notably Siku Quanshu, along with other historical texts. The effectiveness of our method is demonstrated by the experimental results obtained from two test sets. Our results reveal performance gains compared to the baselines employed in the evaluation. Our findings not only showcase the adaptability of the fine-tuned model on this downstream task but also demonstrate the generalization capabilities of the Xunzi model in the domain of ancient Chinese text processing.

2. Related Work

2.1. Sentence Segmentation and Sentence Punctuation

Methods of Chinese sentence segmentation can be primarily classified into rule-based, sequence labeling model-based, and neural network language model-based approaches.

Rule-based methods are not suitable for large-scale processing of ancient texts. In recent years, research has often treated sentence segmentation in ancient Chinese texts as a sequence labeling problem similar to word segmentation. To address the issue of sentence segmentation in ancient texts, researchers have employed Conditional Random Fields (CRF) (Lafferty et al., 2001) for modeling purposes. Also, the combination of LSTM and CRF models (Wang et al., 2019) often yields better results. Wang et al. propose a sentence segmentation method for ancient Chinese texts based on neural network language models (Wang et al., 2016).

Sentence punctuation has a wide range of application scenarios in the field of speech recognition, as the textual sequences generated after recognition often lack punctuation. While neural network methods have achieved considerable success in restoring punctuation in English text, there have been relatively few efforts made to apply these techniques to Chinese punctuation restoration (Zhang et al., 2020), let alone ancient Chinese texts.

2.2. Pre-trained Language Model

Currently, large language models based on the Transformers architecture, such as GPT, T5, and BERT, have achieved state-of-the-art (SOTA) results in various natural language processing tasks. Fine-tuning pre-trained language models on downstream tasks has become a paradigm for handling NLP tasks. Compared to using out-of-the-box pre-trained LLMs (e.g., zero-shot inference), fine-tuning these pretrained LLMs on downstream datasets yields significant performance improvements. The idea behind Domain-Adaptive Pre-training is to adapt the model to a particular domain by exposing it to domain-specific language patterns, terminology, and characteristics during the pre-training phase.

However, as models grow larger, performing full parameter fine-tuning on consumer-grade hardware becomes infeasible. Additionally, storing and deploying individually fine-tuned models for each downstream task becomes highly expensive due to the comparable size of fine-tuned models to the original pre-trained models. Consequently, in recent years, researchers have proposed various parameter-efficient transfer learning methods (Lialin et al., 2023). These methods involve fixing the majority of parameters in the pre-trained model and only adjusting a small subset of parameters to achieve similar effects as full fine-tuning. The adjusted parameters can include both inherent model parameters and additional ones introduced.

3. Method

3.1. Pre-processing

We performed pre-processing on the raw data. Firstly, we detected duplicate sentences in the training data. Most of these are short sentences, such as “其二 (the second)” appearing 349 times and “宋史 (history of the Song dynasty)”appearing 174 times. As these duplicates do not contribute to performance improvement in model training, we retained only one instance of each sentence. In addition, within the training data, there are some texts lacking punctuation annotations, possibly due to annotation oversights or missing original historical records, such as “和君擊築吟請君側耳聽不是更容貌誰能知姓名主人莫稱善坐客何須驚酒酣欲罷奏壯心難自平 (With you, I strike the zither and sing, please listen closely, as appearances may be deceiving and the name of the master remains unknown, so let the seated guest not be startled, for with wine in hand and the music about to end, a strong heart finds it hard to be at peace)”. In order to maintain a high standard of quality in the training data, we decided to remove these sentences which have a length of over 30.

Unlike the conventional paradigm used by previous expert models, the current LLMs primarily employ the “training + context” learning paradigm. As a result, it is necessary to select appropriate prompt templates for each downstream task to help the model recall the knowledge it acquired during training, thus achieving alignment between the downstream and pre-training tasks. The training data is partitioned into fixed-length segments, where the input consists of text sequences with designated punctuation removed, and the output is the original text. The instruction specifies, “Please add punctuation to the following unpunctuated classical Chinese passage without any additional output.” To optimize context token length, no examples are included in the prompt.

3.2. Model

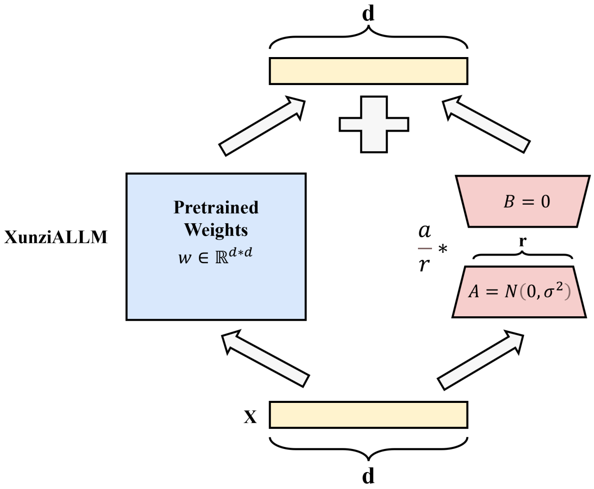


Figure 1: The structure of LoRA model.

We utilized the large language model XunziALLM, which is built upon the Qwen-7B (Bai et al., 2023) model and further pre-trained using corpora consisting of classical Chinese texts. Consequently, XunziALLM possesses extensive knowledge of classical Chinese and various capabilities in processing classical texts. To avoid making full parameter modifications to the original large-scale model, we employed the LoRA method for efficient parameter fine-tuning and supervised training, known as SFT (Supervised Fine-Tuning).

The principle behind the LoRA model involves approximating the incremental updates with low-rank matrices A and B, which are placed alongside the original pre-training matrix. This approximation is used to perform parameter updates efficiently. A large-scale model processes data by mapping it into a high-dimensional space. In fact, when dealing with a specific and narrow task, it may not be necessary to employ such a complex large-scale model. Instead, it might be sufficient to focus on a sub-space range to address the task. We can define the intrinsic rank of the parameter matrix in the sub-space as the rank that achieves a certain level of performance comparable to optimizing the full parameters for the specific problem at hand.

$$\begin{aligned} W_0 + \Delta W &= W_0 + BA \\ B \in \mathbb{R}^{d\times r}, A &\in \mathbb{R}^{r\times k}, r \ll \min(d,k) \end{aligned} \tag{1}$$

Figure 1 shows the structure of LoRA model. As can be seen in Formula 1, the pre-trained weight matrix W_0 can be approximated using a low-rank decomposition to represent the parameter update ΔW. During the training process, the parameters of W_0 are frozen, and only the parameters in A and B are trained. For $h = W_0 x$, the forward propagation process is modified as follows:

$$\begin{aligned} h &= W_0 x + \Delta W x \\ &= W_0 x + BAx \end{aligned} \tag{2}$$

During the training process, the low-rank adaptation matrix amplifies the useful features for downstream tasks, enabling the large-scale model to adapt to sentence punctuation tasks in classical texts.

3.3. Post-processing

Accuracy is a crucial aspect in sentence segmentation and punctuation tasks. Due to the greedy sampling approach employed by large models and the variations in input tokenization, the direct output of the model often contains mistakes. For example, there might be instances where the model overlooks a particular character from the original text or introduces an extra character. To address these issues, we have identified and abstracted most of the possible scenarios and implemented a post-processing step for refining the output generated by the Xunzi model.

As can be seen in Algorithm 1, this post-processing step aims to rectify inaccuracies and inconsistencies in the punctuation predictions by considering the specific context and linguistic rules. As a result, we enhance the reliability and coherence of the model's output, ensuring that it aligns with the intended punctuation patterns in practical usage scenarios.

Algorithm 1 Post-process

```
Input: original sentence $s_1$ and sentence after punctuation $s_2$
  $s_3 \leftarrow move\_punctuation(s_2)$
  if $s_1$ == $s_3$ then
    return $s_2$
  else
    if $len(s_3)$ == $len(s_1)$ then
      $IDs \leftarrow s_3[id] \neq s_1[id]$
      for $id$ in $IDs$ do
        $s_3[id] \leftarrow s_1[id]$
      end for
    else
      $b_{s_1} \leftarrow 0, e_{s_1} \leftarrow len(s_1) - 1$
      $b_{s_3} \leftarrow 0, e_{s_3} \leftarrow len(s_3) - 1$
      while $s_1[b_{s_1}] == s_3[b_{s_3}]$ do
        $b_{s_1} \leftarrow b_{s_1} + 1$
        $b_{s_3} \leftarrow b_{s_3} + 1$
      end while
      while $s_1[e_{s_1}] == s_3[e_{s_3}]$ do
        $e_{s_1} \leftarrow e_{s_1} - 1$
        $e_{s_3} \leftarrow e_{s_3} - 1$
      end while
      $s_3[b_{s_3} : e_{s_3}] \leftarrow s_1[b_{s_1} : e_{s_1}]$
      $s_2 \leftarrow restore\_punction(s_3)$
    end if
  end if
Output: $s_2$
```

4. Experiments

4.1. Dataset

We used the training dataset released by EvaHan2024, which consists of texts from classical sources. The corpus of ancient Chinese classical texts demonstrates a diachronic nature, encompassing a vast time span of thousands of years and encompassing the four traditional categories of Chinese canonical texts, namely *Jing (经)*, *Shi (史)*, *Zi (子)*, and *Ji (集)*. We conducted a statistical analysis on the occurrence of punctuation marks in the training text, as shown in Table 2. It is worth mentioning that the corner brackets 【 】appeared 53818 times. Since they are not within the scope of sentence segmentation and punctuation in this context, we can treat them as two special Chinese characters.

There are two test datasets. Test A includes approximately 50000 characters of Ancient Chinese texts and comes from different sources. Test B mainly comes from the book *Zuo Zhuan*.

4.2. Metric

Precision (P), Recall (R), and F1 Score are employed as evaluation metrics for all experiments, with the results being expressed in percentages.

Task (Test A)	**Seg**			**Punc**		
	Precision	Recall	F1	Precision	Recall	F1
Xunzi-Qianwen-7B-CHAT	90.53	66.12	76.42	73.52	52.22	61.06
ChatGPT 3.5	83.81	59.85	69.83	63.90	43.88	52.03
Our system	**90.80**	**76.34**	**82.94**	**77.75**	**63.85**	**70.12**
Task (Test B)	**Seg**			**Punc**		
	Precision	Recall	F1	Precision	Recall	F1
Xunzi-Qianwen-7B-CHAT	95.28	87.17	91.04	79.25	72.09	75.50
Our system	**95.98**	**90.54**	**93.18**	**85.08**	**79.93**	**82.43**

Table 1: Experimental results on two tests.

Punctuation	Name	Count
，	Comma	1879220
。	Period	954948
、	Slight-pause	126394
：	Colon	163968
；	Semicolon	55256
？	Question	73067
！	Exclamation	45623
“ ”	Double Quotes	240176
‘ ’	Single Quotes	10036
《 》	Book Title Mark	120558

Table 2: Statistics of punctuation marks in the training text

4.3. Implementation Details

We utilized the latest XunziALLM as the base model. A complete three-round training using LoRA was conducted on a device with a Nvidia A100 40G. Each training session, which lasted approximately 20 hours, was conducted on LLaMA Factory (Zheng et al., 2024), an integrated large-scale model training platform. During training, the learning rate is set to $4e-5$, and the LR scheduler is $cosine$. The other important hyperparameters are listed in Table 3.

Hyperparameters	Value
Learning rate	4e-5
LR scheduler	cosine
Warmup steps	200
LoRA rank	8
LoRA Alpha	32
LoRA modules	all

Table 3: Hyperparameters of training

4.4. Results

The results are shown in Table 1. Compared to baselines, our approach achieved comprehensive improvements in sentence segmentation and punctuation tasks. In terms of different test tasks, our method performs relatively well on test B, specifically the *Zuo Zhuan* dataset. Possibly because it has been specifically tailored to understand the stylistic consistency and historical context of classical Chinese texts, with refined preprocessing and post-processing steps that effectively capture the unique linguistic patterns and nuances present in this historical narrative. This validates the effectiveness of the additional layers we designed and demonstrates the advantages of the Xunzi model in processing classical Chinese texts.

5. Conclusion

In this paper, we described the method for tasks in EvaHan2024 using the LoRA approach in the context of ancient Chinese text processing. Our focus has been on sentence segmentation and punctuation. By leveraging the training dataset and building upon Xunzi model, we demonstrated significant improvements over baselines in these tasks. Our experimental results on the test sets, particularly the Zuo Zhuan dataset (test B), validate the effectiveness of our method and showcases its robustness, accuracy, and generalization capabilities.

However, the method may have some limitations. For instance, it relies on a series of predefined rules to correct the model's output, which may not cover all types of errors and may not be flexible enough when dealing with complex or atypical texts.

Automated sentence segmentation and punctuation play a vital role in promoting the study and preservation of ancient books, as well as the inheritance of Chinese culture. With further advancements and refinements in this area, we can contribute to the broader accessibility and understanding of classical Chinese literature for scholars and readers worldwide.

6. References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Icml*, volume 1, page 3. Williamstown, MA.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*.

Boli Wang, Xiaodong Shi, Zhixing Tan, Yidong Chen, and Weili Wang. 2016. A sentence segmentation method for ancient chinese texts based on nnlm. In *Chinese Lexical Semantics*, pages 387–396, Cham. Springer International Publishing.

Hongbin Wang, Haibing Wei, Jianyi Guo, and Liang Cheng. 2019. Ancient chinese sentence segmentation based on bidirectional lstm+ crf model. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 23(4):719–725.

Zhe Zhang, Jie Liu, Lihua Chi, and Xinhai Chen. 2020. Word-level bert-cnn-rnn model for chinese punctuation restoration. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pages 1629–1633.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

Sentence Segmentation and Punctuation for Ancient Books based on Supervised In-context Training

Shiquan Wang, Weiwei Fu, Mengxiang Li, Zhongjiang He,
Yongxiang Li, Ruiyu Fang, Li Guan, Shuangyong Song
China Telecom Corporation Ltd. AI Technology Company
{wangsq23, fuweiwei, hezj, liyx25, guanl, fangry, songshy}@chinatelecom.cn
limengx@126.com

Abstract

This paper describes the participation of team "TeleAI" in the third International ancient chinese Language Information Processing Evaluation (EvaHan2024). The competition comprises a joint task of sentence segmentation and punctuation, categorized into open and closed tracks based on the models and data used. In the final evaluation, our system achieved significantly better results than the baseline. Specifically, in the closed-track sentence segmentation task, we obtained an F1 score of 0.8885, while in the sentence punctuation task, we achieved an F1 score of 0.7129.

Keywords: sentence segmentation, sentence punctuation, in-context learning

1. Introduction

The tasks of sentence segmentation and punctuation in Chinese ancient texts are significant challenges and hold great importance in the field of Natural Language Processing (NLP). In ancient chinese texts, sentences are often written without explicit punctuation, making it difficult for modern readers and NLP systems to accurately interpret the text's structure and meaning. Sentence segmentation involves identifying boundaries between sentences, which is crucial for tasks such as text comprehension, information extraction, and machine translation. Furthermore, restoring missing or ambiguous punctuation marks is essential for improving the readability and understanding of ancient chinese texts. These tasks not only contribute to the preservation and analysis of historical texts but also serve as fundamental building blocks for various NLP applications, including language understanding, generation, and translation. Thus, addressing the challenges of sentence segmentation and punctuation restoration in ancient chinese is essential for advancing research in NLP and facilitating cross-temporal communication and understanding.

Our submitted system adopts a two-stage strategy to improve the performance of the model on ancient chinese sentence segmentation and punctuation restoration tasks. In the first stage, we enhance the performance of the XunziALLM base model through Supervised In-context Training. In the second stage, we improve the model's accuracy in discerning unreliable punctuation by employing a greedy character correction approach and a voting strategy. Our final submission achieved an F1 score of 0.8885 in the sentence segmentation task and 0.7129 in the joint task of sentence segmentation and punctuation.

2. Related Work

The absence of punctuation and sentence breaks in ancient chinese has been a longstanding cultural convention. However, the lack of sentence breaks poses a challenge for modern individuals in learning and utilizing ancient chinese. Manual sentence segmentation requires a clear understanding of semantics, grammar, rhythm, and indicative words, and consumes a significant amount of time and effort. To better understand and study the grammatical structure, logical relationships, and expression methods of ancient texts, as well as to grasp the semantics and rhythm of sentences, a large number of researchers are exploring the joint task of automatic sentence segmentation and punctuation restoration in ancient texts using natural language processing techniques.

Early punctuation restoration tasks were predominantly based on rules along with LSTM, CNN, and other deep learning models. Tilk and Alumäe (2015) propose an LSTM-based punctuation restoration approach, first learning text features on a large text corpus, and then utilizing text features and prosodic features to predict punctuation marks on a small-scale corpus. Che et al. (2016) initially transform the text into a long word sequence, treating the punctuation restoration task as a sequence classification problem, and utilize Deep Neural Networks (DNN), Convolutional Neural Networks (CNN-A), and Double-layer Convolutional Neural Networks (CNN-2A) to predict punctuation marks. Cheng and Li (2020) proposed a method based on BiLSTM+CRF to achieve joint annotation of sentence segmentation and lexical analysis in Classical Chinese. They validated the effectiveness of this approach on four different test sets from different periods. Kim (2019) introduced a recurrent neural network model based on hierarchical multi-head attention. This model employs hierarchical attention to allow each layer to learn different contexts from various perspectives.

With the advancement of deep learning technologies, punctuation restoration methods based on transformers and pretrained language models have achieved significant success. Wang et al. (2018) framed punctuation restoration as a translation task, where the model takes unpunctuated sequences as input and produces sequences of punctuation marks and labels as output. This approach leverages Transformer networks based on self-attention mechanisms to extract hidden features. Wang et al. (2022) utilized validated high-quality corpora of the entire texts from the "Siku Quanshu" as the training set to construct SikuBERT and SikuRoBERTa for ancient chinese intelligent processing tasks. They validated the performance of these models across multiple ancient chinese tasks.

With the remarkable achievements of large language models (LLMs) in various fields of natural language processing, there has been a growing emphasis on integrating LLMs with classical literature processing to advance intelligent research on ancient texts. In this context, Nanjing Agricultural University has introduced the XunziALLM, aiming to facilitate the intelligent processing of classical texts. XunziALLM has demonstrated significant potential across multiple downstream tasks related to ancient texts.

3. Method

3.1. Supervised In-context Training

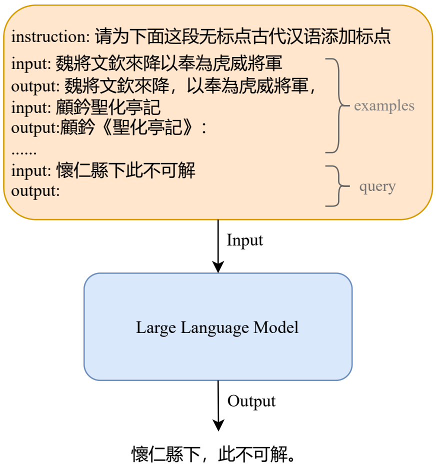


Figure 1: Illustration of in-context learning

With the scaling of model size and corpus size, large language models (LLMs) demonstrate an in-context learning (ICL) ability, wherein they learn from a few examples in the context. Numerous studies have indicated that LLMs can effectively perform a variety of complex tasks through ICL. Figure 1 provides an illustrative example depicting how language models make decisions using ICL. In essence, the model estimates the likelihood of potential answers conditioned on the demonstration, leveraging a well-trained language model.

Formally, given a query input text x and a set of candidate answers $Y = \{y_1, \ldots, y_m\}$, a pretrained language model M selects the candidate answer with the maximum score as the prediction, conditioning on a demonstration set C. The set C comprises an optional task instruction I and K demonstration examples, thus $C = \{I, s(x_1, y_1), \ldots, s(x_k, y_k)\}$ or $C = \{s(x_1, y_1), \ldots, s(x_k, y_k)\}$, where $s(x_k, y_k, I)$ represents an example written in natural language texts according to the task. The likelihood of a candidate answer y_j can be represented by a scoring function f of the entire input sequence with the model M :

$$P(y_j|x) \triangleq f_M(y_j, C, x) \tag{1}$$

The final predicted label $\hat{y}$ is the candidate answer with the highest probability:

$$y = \arg\max_{y_j \in Y} P(y_j|x) \tag{2}$$

While LLMs have demonstrated promising ICL capability, several studies also suggest that this capability can be further enhanced through a continual training stage between pretraining and ICL inference (Wei et al., 2023; Chen et al., 2022). Therefore, we enhance the ICL capability of LLMs by constructing context-learning instruction training data and eliminating the gap between pretraining tasks and downstream ICL tasks through supervised instruction fine-tuning. Specifically, we utilize a differential selection method to choose example data in ICL and construct supervised ICL training data, followed by training XunziALLM based on the supervised ICL data.

3.2. Character Correction and Voting Strategy

During the inference process, we observed discrepancies between some predictions of the Large Language Model (LLM) and the input text at the character level. To ensure consistency between the model's predictions and the original text, we propose a greedy character correction algorithm, as shown in Algorithm 1. This algorithm sequentially examines the characters in the predicted and original texts. If a character is a punctuation mark, it is directly appended to the result string. Otherwise, each character in the original and predicted texts is compared, and corresponding operations, such

Algorithm 1 Greedy Character Correction Algorithm

```
Require: Original text $original\_text$, Predicted text $predicted\_text$
Ensure: Text with restored punctuation $res$
Initialize empty string $res$
Initialize indices $i \leftarrow 0, j \leftarrow 0, max\_try \leftarrow 0$
while $i < $ length($original\_text$) and $j < $ length($predicted\_text$) do
    $max\_try \leftarrow max\_try + 1$
    if $max\_try > 100000$ then
        break
    end if
    if $predicted\_text[j]$ is a punctuation mark then
        Append $predicted\_text[j]$ to $res$
        $j \leftarrow j + 1$
        continue
    end if
    if $original\_text[i] = predicted\_text[j]$ then
        Append $original\_text[i]$ to $res$
        $i \leftarrow i + 1$
        $j \leftarrow j + 1$
        continue
    end if
    $k \leftarrow j$
    while $predicted\_text[k+1]$ is a punctuation mark and $(k + 1) <$ length($predicted\_text$) do
        $k \leftarrow k + 1$
    end while
    if $original\_text[i + 1] = predicted\_text[k + 1]$ then
        Append $original\_text[i]$ to $res$
        $i \leftarrow i + 1$
        $j \leftarrow j + 1$
        continue
    end if
    if $original\_text[i + 1] = predicted\_text[j]$ then
        Append $original\_text[i]$ to $res$
        $i \leftarrow i + 1$
        continue
    end if
    if $original\_text[i] = predicted\_text[k + 1]$ then
        $j \leftarrow j + 1$
        continue
    end if
end while
if $j <$ length($predicted\_text$) then
    Append remaining characters of $predicted\_text$ to $res$
end if
return $res$
```

as replacement, deletion, or addition of characters, are performed based on their equality or inequality.

Simultaneously, we observed discrepancies in the predictions of different models for the same input, reflecting variations in the models' confidence levels regarding candidate entities. To leverage the advantages of different models, mitigate the limitations of individual models, and enhance overall predictive performance, we initially retained predictions from multiple models across different iterations. Subsequently, we employed a voting method to obtain the final prediction result.

4. Experiments

This section will introduce the experimental aspects involved in our participation in this evaluation task, primarily encompassing three parts: data preprocessing, experimental parameter settings, and experimental results and analysis.

4.1. Data Preprocessing

The EvaHan2024 dataset comprises texts sourced from classical literature, especially the Siku Quanshu (Four Treasuries) and other historical texts. Constructed through initial label predictions by models and subsequent human expert corrections, the original training set consists of 254,360 data points, with 412 data points in the test set. Through rule-based filtering, we selected 126,372 high-quality data points from the training set, which were then redivided into training and validation sets in a 9:1 ratio. All subsequent comparative experiments were conducted based on this redivided training and validation set.

The evaluation in this assessment involves two tasks: sentence segmentation and sentence Punctuation. Sentence segmentation is the process of converting Chinese text into a sequence of sentences, with each sentence separated by a single space. Additionally, sentence punctuation involves correctly placing punctuation marks at the end of each sentence. In many Chinese language processing systems, these two tasks, sentence segmentation and punctuation, are typically addressed together. Therefore, we developed a set of evaluation scripts for offline assessment of this joint task, calculating precision, recall, and F1 scores. Based on the offline evaluation results, we selected the optimal outcome as the final submission version. The scores on the validation sets in subsections 4.2 and 4.3 are all computed based on this offline evaluation script.

4.2. Experiment1: Supervised Fine-tuning

In pursuit of identifying the most suitable XunziALLM base model for handling sentence segmentation and punctuation restoration tasks in Classical Chinese, we conducted experiments on the re-partitioned EvaHan2024 dataset. The experimental findings are presented below.

Table 1 showcases the performance of various XunziALLM base models on the EvaHan2024 dataset. Precision, Recall, and F1 metrics denote

Model	Precision	Recall	F1	ER
Xunzi-Qwen	0.7517	0.6961	0.7228	0.19
Xunzi-Qwen-CHAT	0.7573	0.7079	0.7318	0.32
Xunzi-GLM	0.7548	0.7365	0.7455	0.06
Xunzi-Baichuan	**0.7759**	**0.7588**	**0.7672**	**0.04**

Table 1: Experimental Results of XunziALLM Models on the EvaHan2024 Dataset

the highest scores achieved by different XunziALLM base models in the joint task of sentence segmentation and punctuation restoration, while ER represents the proportion of character inconsistencies between model predictions and the original texts. The experimental results reveal that Xunzi-Baichuan attained the highest F1 score on the EvaHan2024 dataset, accompanied by the lowest proportion of character discrepancies between predicted results and the original texts. Consequently, for subsequent experiments, we elected to utilize this model as the primary base model.

4.3. Experiment2: Supervised In-context Training

While pre-trained language models have demonstrated initial capabilities in In Context Learning (ICL), there remains a certain gap between their pretrained objectives and downstream ICL tasks. To fully harness the potential of XunziALLM in context learning, we employed a differential selection approach to curate sample data suitable for ICL, thereby constructing a supervised ICL training dataset. Subsequently, we trained XunziALLM based on this supervised ICL dataset. The experimental results are shown in Table 2.

Model	Precision	Recall	F1	ER
Xunzi-Qwen	0.7640	0.7250	0.7465	0.09
Xunzi-Qwen-CHAT	0.7687	0.7399	0.7540	0.35
Xunzi-GLM	0.7862	0.7639	0.7749	**0.04**
Xunzi-Baichuan	**0.8013**	**0.7892**	**0.7952**	**0.04**

Table 2: Experimental Results of XunziALLM Models on the EvaHan2024$_{\text{ICL}}$ Dataset

Table 2 illustrates the performance of XunziALLM on the EvaHan2024 dataset after supervised ICL training. The results indicate that supervised fine-tuning with ICL supervision enhances the ability of LLMs to learn from context during inference, thereby improving XunziALLM's performance on sentence segmentation and punctuation restoration tasks in classical chinese. Simultaneously, it can be observed that supervised in-context training outperforms direct supervised fine-tuning.

4.4. Experiment3: Online Submission

Table 3 presents the experimental results of our system compared with the baseline model (Xunzi-Qwen-7B-Chat) on Test Set A, while Table 4 presents the experimental results of our system compared with the baseline model on Test Set B. In the joint task of sentence segmentation and punctuation, our system achieved a relative improvement of 16.75% on Test Set A and 9.13% on Test Set B compared to the baseline model. These experimental results demonstrate the effectiveness of our proposed method, indicating that supervised in-context training can enhance the performance of models in sentence segmentation and punctuation tasks for ancient texts.

Model	Task	Precision	Recall	F1
Xunzi-Qwen-7B-Chat	Seg	0.9053	0.6612	0.7642
	Punc	0.7352	0.5222	0.6106
Our System	Seg	**0.9170**	**0.8671**	**0.8885**
	Punc	**0.7433**	**0.6848**	**0.7129**

Table 3: Experimental Results on the Test Set A Compared with Baseline Model.

Model	Task	Precision	Recall	F1
Xunzi-Qwen-7B-Chat	Seg	0.9528	0.8717	0.79104
	Punc	0.7925	0.7209	0.7550
Our System	Seg	**0.9632**	**0.9146**	**0.9383**
	Punc	**0.8599**	**0.7910**	**0.8240**

Table 4: Experimental Results on the Test Set B Compared with Baseline Model.

5. Conclusion

In this paper, we describe our submission system for the EvaHan2024 shared task. We present our solution in two stages: (a) Supervised In-context Training and (b) Character Correction and Voting. In the final evaluation, our system achieved outstanding results in the closed track, with a final F1 score of 0.7129 on Test Set A and 0.8240 on Test Set B.

6. References

Xiaoyin Che, Cheng Wang, Haojin Yang, and Christoph Meinel. 2016. Punctuation prediction for unsegmented transcript based on word vector. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 654–658.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022. Improving in-context few-shot learning via self-supervised training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3558–3573.

Qian Chen, Wen Wang, Mengzhe Chen, and Qinglin Zhang. 2021. Discriminative self-training for punctuation prediction. *arXiv preprint arXiv:2104.10339*.

Ning Cheng and Bin Li. 2020. A joint model of automatic sentence segmentation and lexical analysis for ancient chinese based on bilstm-crf model.

Maury Courtland, Adam Faulkner, and Gayle McElvain. 2020. Efficient automatic punctuation restoration using bidirectional transformers with robust inference. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 272–279.

Seokhwan Kim. 2019. Deep recurrent neural networks with layer-wise multi-head attentions for punctuation restoration. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7280–7284. IEEE.

Ottokar Tilk and Tanel Alumäe. 2015. Lstm for punctuation restoration in speech transcripts. In *Interspeech*, pages 683–687.

Dongbo Wang, Chang Liu, Zihe Zhu, Jiangfeng Liu, Haotian Hu, Si Shen, and Bin Li. 2022. Construction and application of pre-trained models of siku quanshu in orientation to digital humanities. *Library tribune*, 42(06):31–43.

Feng Wang, Wei Chen, Zhen Yang, and Bo Xu. 2018. Self-attention based network for punctuation restoration. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2803–2808. IEEE.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, et al. 2023. Symbol tuning improves in-context learning in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 968–979.

SPEADO: Segmentation and Punctuation for Ancient Chinese Texts via Example Augmentation and Decoding Optimization

Xia Tian[1], **Yu Kai**[2], **Yu Qianrong**[1], **Peng Xinran**[1]
[1]School of Information Resource Management, Renmin University of China, Beijing, China
[2]Shanghai Midu Technology Co., Ltd., Shanghai, China
[1]{xiat, yuqianrong77, pengxinran166}@ruc.edu.cn
[2]yukai@midu.com

Abstract

The SPEADO model for sentence segmentation and punctuation tasks in ancient Chinese texts is proposed, which incorporates text chunking and MinHash indexing techniques to realise example argumentation. Additionally, decoding optimization strategies are introduced to direct the attention of the LLM model towards punctuation errors and address the issue of uncontrollable output. Experimental results show that the F_1 score of the proposed method exceeds the baseline model by 14.18%, indicating a significant improvement in performance.

Keywords: Sentence segmentation and punctuation, Ancient Chinese texts, Large Language Models

1. Introduction

Ancient texts, a crucial component of Chinese culture, are abundant in historical, cultural, and ideological value. However, their distinctive ancient writing style often lacks explicit sentence breaks and punctuation, rendering them difficult to read and comprehend. While traditional manual annotation methods can provide assistance, the vast quantity of ancient Chinese texts makes manual processing inefficient and costly, limiting digital processing and large-scale research efforts. Fortunately, with the advancements in Large Language Model (LLM) technologies, it has become more feasible to efficiently tackle this challenge.

This task can be regarded as a generation task, involving the conversion of unpunctuated sentences into punctuated ones. LLMs, such as XunziALLM, have demonstrated remarkable fundamental capabilities in this regard. We propose an augmentation method inspired by human learning through examples, coupled with decoding strategies to enhance task focus and output control. Fine-tuning with LoRA enables our SPEADO model to learn the skill of punctuating ancient Chinese texts, significantly improving performance over baseline models.

2. Related Work

The sentence segmentation and punctuation tasks are crucial for parsing the meaning of Ancient Chinese texts. Research on automated annotation methods for these tasks can be categorized into several stages: rule-based, statistical, and deep learning approaches. Early attempts to these tasks primarily relied on rule-based systems (Huang and Hou, 2008). While effective in some cases, rule-based approaches often struggled with ambiguous syntactic structures and variations in writing styles. Moreover, maintaining and updating rule sets proved to be labor-intensive and prone to errors. Therefore, researchers started exploring natural language statistical modeling, particularly the development of N-gram models that capitalized on contextual features to predict sentence boundaries (Cheng et al., 2007).

With the development of the field of deep learning and the advancement of sentence segmentation and punctuation tasks in Acient Chinese, models such as BERT, LSTM/BiLSTM, and CRF (Yu et al., 2019; Wang et al., 2021) have been proven to exhibit strong performance in there.Subsequently, researchers have shifted their focus towards optimizing these network architecture.

Some researchers have focused on optimizing pre-trained models and, based on large-scale Classical Chinese datasets, have respectively trained pre-trained models tailored for Classical Chinese, namely BERT_guwen and SikuBERT. Some scholars have incorporated fine-grained textual knowledge and adjusted the model structure using CNN and BiLSTM, proposing the BBiCC-EK (BBiC-CNN-External Knowledge) model (Li et al., 2023). Moreover, Considering that separating punctuation and sentence segmentation in classical texts into two sequential tasks may lead to error propagation, some studies treat the segmentation and punctuation of ancient texts as a joint task (Yuan et al., 2022).

Notable Chinese LLMs include Baidu's Ernie (Yu et al., 2021) and Alibaba Cloud's Qwen (Jinze et al., 2023), demonstrating excellent language understanding and generation abilities. For Acient Chinese, Nanjing Agricultural University and Zhonghua Book Company's joint efforts have produced a series of LLMs specialized in processing classical texts, named as the XunziALLM. These models exhibit impressive performance in handling Ancient Chinese textual information. Leveraging XunziALLM as base model and optimizing it for the joint task like punctuation and sentence segmentation in classical texts seems like a promising choice.

3. Method

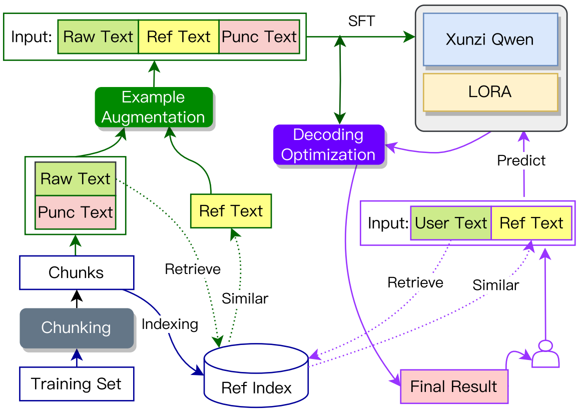


Figure 1: Architecture of SPEADO.

Recognizing that the punctuation in ancient Chinese texts encodes crucial information for sentence segmentation, we have merged the tasks of automatic sentence segmentation and punctuation, introducing an integrated approach named SPEADO. SPEADO, an acronym for sentence segmentation and punctuation via example augmentation and decoding optimization, offers a comprehensive solution to the challenges posed by these tasks. As depicted in Figure 1, SPEADO comprises three core modules: text chunking, example augmentation, and decoding optimization.

3.1. Chunking Process

Because the lengths of the samples in the training data vary significantly, directly utilizing each row as a standalone sample for input into the training network can result in truncation issues and hinder the training speed. To mitigate this issue, we have employed a sliding window mechanism that divides the training dataset into chunks, thereby enabling the generation of additional training samples.

As depicted in Figure 2, let us consider the raw text X comprising of m sentences, denoted as $X = x_1, \cdots, x_m$. We proceed to transform X into a series of length-constrained chunks, designated as $C = \{c_1, c_2, ..., c_n\}$. In the process of conversion, we traverse X from the left to the right, iteratively generating each chunk c_i in the following manner:

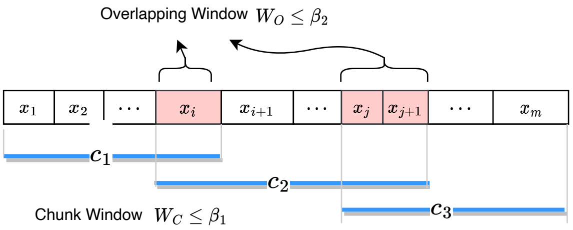


Figure 2: Illustration of Overlapping Chunks.

1. The chunk c_i starts at x_a and ends at some x_b, where $a \leq b \leq m$, and the initial value of a is 1;
2. Find the largest value b that satisfies $\sum_{k=a}^{b} len(x_k) \leq \beta_1$;
3. Set the chunk $c_i = \{x_a, x_{a+1}, \cdots, x_b\}$;
4. Starting from x_b, backtrack to find the smallest value $c \geq 1$ that satisfies $\sum_{k=c}^{b} len(x_k) \leq \beta_2$;
5. Set $a = b$, repeat step 1, and obtain all valid chunks in turn.

Here, β_1 represents the maximum value for the window size of a text chunk, while β_2 denotes the maximum value for the overlapping window.

3.2. Example Augmentation

When humans tackle tasks, the provision of pertinent reference information greatly aids in their resolution. Drawing inspiration from this, we incorporate correct reference examples with the original text during the training process of our model for automatic punctuation of ancient texts. This allows LLM to refer to relevant information to better perform the punctuation task.

To achieve this, we pre-construct a MinHash index for the text chunks obtained in the previous step. The utilization of MinHash, rather than text embedding techniques, stems from the fact that semantic embedding models exhibit limited effectiveness in comprehending ancient Chinese texts. Consequently, MinHash is more adept at retrieving and matching character similarity. After performing the MinHash operation, a reference index database (i.e., Ref Index) is created for the text chunks, enabling us to retrieve examples for reference purposes.

For a manually punctuated text chunk c, we establish the punctuated text as the gold standard and proceed to strip it of all punctuation, yielding the raw text that requires punctuation prediction. Subsequently, we compute the MinHash value of the raw text and utilize it to retrieve a similar text from the refdb, designated as the reference text. The raw text, reference text, and the original punctuated text are then merged according to a prescribed prompt template, culminating in a comprehensive training data input tailored for supervised fine-tuning within LLM.

3.3. Decoding Optimization

After fine-tuning, the LLM still demonstrates unpredictable behavior during prediction, such as generating characters that are neither punctuation nor original text, and reproducing entire sentences without any modification. To tackle this issue, we have incorporated three types of optimization techniques during the model decoding process.

Firstly, we refined the loss function during training to prioritize punctuation errors. Whenever the punctuation placement in the output is inaccurate, we enhance the original loss value using a factor, λ, set to 0.05 in our experiments.

Secondly, during prediction, we imposed a decoding constraint that confines the next predicted character to either the original input character or punctuation marks. Subsequently, we selected the character with the highest logits value from this constrained set as the final prediction, effectively addressing issues pertaining to inconsistent output characters.

Lastly, we utilized a voting mechanism for unchanged sentences after prediction. We trained three models using LoRA fine-tuning based on the Xunzi-Qwen-7B. These included SPEADO-A (standard LoRA fine-tuning), SPEADO-B (with example augmentation), and SPEADO-C (with loss adjustment). These three models form the expert model, which votes on unmodified sentences and re-selects the prediction results.

4. Experiments

4.1. Data and Evaluation Metrics

We employed the dataset provided by the EvaHan2024 organizers for both training and testing. The training set included 10 million characters extracted from the Complete Library of Four Branches. The test sets comprise A and B, where the former refers to the data released initially, and the latter refers to the Zuozhuan data released the second time.

In the model validation phase, we randomly sampled 10,000 lines of text without replacement from the training set to create a validation set, reserving the remaining data for model training, to observe the varying impacts of different factors on the model. In the final stage, we utilized all the data as the training set to predict on the test sets. The prediction results were then submitted to the organizers for metric calculations. Table 1 presents the detailed statistical information of the dataset.

Dataset	Samples	Max Len	Avg Len
Training Set	254,360	29,907	93
Validation Set	10,000	3,546	116
Test Set - A	412	1,569	122
Test Set - B	3,319	656	59

Table 1: Statistics of EvaHan2024 dataset.

Table 1 reveals a considerable disparity in the average length of samples, with the longest sentence in the training set spanning 29,907 characters while averaging just 93. To mitigate this and enhance our dataset, we divided the raw text into chunks using parameters $\beta_1 = 256$ and $\beta_2 = 128$. This approach not only augmented our sample size but also addressed the challenge of excessively long inputs.

Following the convention of Seg and Punc tagging, we use Precision (P), Recall (R), and F1 Score as the evaluation metrics for all experiments. All the results are presented in percentages (%).

4.2. Implementation Details

For all experiments, we utilize the Xunzi-Qwen-7B as the backbone, employing a learning rate of $1 \times 10E - 5$ for the PLM. We adopt AdamW as the optimizer and WarmupDecayLR as the scheduler. Each GPU is assigned a micro-batch size of 2 for training. All our experiments are conducted on A100 GPUs, requiring approximately 60GiB of GPU memory and taking around 12 hours to achieve optimal performance.

4.3. Results

We compared five different methods on the validation set, and the results are presented in Table 2.

In Table 2, M_1 directly utilizes the Xunzi-Qwen-7B-CHAT model, revealing that the instructed LLM already possesses a certain level of ability in segmentation and punctuation task. M_2 fine-tunes the Xunzi-Qwen-7B base model using LoRA, significantly improving the performance compared to the chat model, emphasizing the necessity of secondary training for specific tasks. M_3 corresponds to the results after fine-tuning with the Xunzi-Baichuan-7B base model, aimed at verifying the differences between various base models. The data indicates that Xunzi-Qwen-7B slightly outperforms Xunzi-Baichuan-7B in this task, leading to our choice of Xunzi-Qwen-7B as our base model.

M_4 investigates the impact of weighting the loss related to punctuation positions during training. We observed a slight decline in certain metrics. Upon analysis, we found that the model's sensitivity to

ID	Base Model	Tuning Method	Seg			Punc		
			P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
M_1	Xunzi-Qwen-7B-CHAT	—	69.60	76.61	72.94	55.02	60.56	57.65
M_2	Xunzi-Qwen-7B	LoRA	75.10	81.57	78.20	62.22	67.57	64.78
M_3	Xunzi-Baichuan-7B	LoRA	75.28	80.80	77.95	62.33	66.90	64.54
M_4	Xunzi-Qwen-7B	LoRA	74.35	80.80	77.40	61.17	66.48	63.71
M_5	Xunzi-Qwen-7B	LoRA	**75.93**	**81.90**	**78.80**	**62.82**	**67.75**	**65.19**

Table 2: Comparison of different methods on the validation set.

Test Sets	Model	Seg			Punc		
		P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
Test Set - A	Xunzi-Qwen-7B-CHAT	90.53	66.12	76.42	73.52	52.22	61.06
	ChatGPT 3.5	83.81	59.85	69.83	63.90	43.88	52.03
	SPEADO	**90.99**	**85.99**	**88.42**	**78.75**	**72.02**	**75.24**
Test Set - B	Xunzi-Qwen-7B-CHAT	95.28	87.17	91.04	79.25	72.09	75.50
	SPEADO	**95.05**	**90.05**	**92.48**	**82.92**	**77.30**	**80.01**

Table 3: Comparison of various methods on the test sets. The asterisk (*) signifies that the EvaHan2024 organizer supplied the test results.

punctuation positions increased, resulting in more precise and nuanced punctuation usage. However, this enhanced sensitivity also led to the generation of redundant punctuation marks. Further exploration is needed to retain the model's stronger correction abilities while suppressing excessive modifications. M_5 introduces an example augmentation technique, which enables the model to better tackle the task by providing similar reference examples. This method demonstrated significant effectiveness.

During the testing phase, we introduced a comprehensive decoding enhancement strategy, employing M_2, M_4 and M_5 as expert model A, B, and C, respectively, to form the complete SPEADO model for prediction. As shown in Table 3, it is evident that SPEADO significantly improves the effectiveness of the tasks compared to the baseline model, Xunzi-Qwen-7B-CHAT.

5. Conclusion

Drawing from the previously mentioned research, it becomes apparent that the combination of example augmentation and decoding optimization can greatly enhance the abilities of LLMs in understanding and addressing tasks related to sentence segmentation and punctuation in ancient Chinese texts. This approach effectively tackles the challenge of uncontrollable output that is typically inherent in LLMs. Furthermore, training on the LLM-base has proven to be a more efficient and targeted means of achieving specific task objectives, surpassing the performance of the LLM-chat version.

6. Acknowledgements

This research is supported by the the National Social Science Fund of China (22BTQ068, 20&ZD260) and the Research Funds of Renmin University of China (22XNQT45).

7. References

Tianying Cheng, Rong Cheng, Lulu Pan, Hongjun Li, and Zhonghua Yu. 2007. Archaic chinese punctuating sentences based on context n-gram model. *Computer Engineering*, (03):192–193+196.

Jiannian Huang and Hanqing Hou. 2008. Review and trend of researches on ancient chinese character information processing. *Journal of Chinese Information Processing*, 22(4):31–38.

Bai Jinze, Bai Shuai, Chu Yunfei, and et al. 2023. Qwen technical report. Technical report.

Peiqi Li, Hao Wang, Qiutong Ren, and Tao Fan. 2023. Study of antiquarian punctuation recognition methods incorporating semantic enhancement with structural properties. *Journal of the China Society for Scientific and Technical Information*, (02):150–163.

Qian Wang, Dongbo Wang, Bing Li, and Chao Xu. 2021. Deep learning based automatic sentence segmentation and punctuation model for massive classical chinese literature. *Data Analysis and Knowledge Discovery*, (03):25–34.

Jingsong Yu, Yi Wei, and Yongwei Zhang. 2019. Automatic ancient chinese texts segmentation based on bert. *Journal of Chinese Information Processing*, (11):57–63.

Sun Yu, Wang Shuohuan, Feng Shikun, and et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint*.

Yiguo Yuan, Bing Li, Minxuan Feng, Sheng He, and Dongbo Wang. 2022. A joint model of automatic sentence segmentation and punctuation for ancient classical texts based on deep learning. *Library and Information Service*, (22):134–141.

Ancient Chinese Punctuation via In-Context Learning

Jie Huang
NANJING UNIVERSITY
Jiangsu, China
huangjie@smail.nju.edu.cn

Abstract

EvaHan2024 focuses on sentence punctuation in ancient Chinese. Xunzi large language base model, which is specifically trained for ancient Chinese processing, is advised in the campaign. In general, we adopted the in-context learning (ICL) paradigm for this task and designed a post-processing scheme to ensure the standardability of final results. When constructing ICL prompts, we did feature extraction by LLM QA and selected demonstrations based on non-parametric metrics. We used Xunzi in two stages and neither did further training, so the model was generic and other fundamental abilities remained unaffected. Moreover, newly acquired training data can be directly utilized after identical feature extraction, showcasing the scalability of our system. As for the result, we achieved an F1-score of 67.7% on a complex test dataset consisting of multiple types of documents and 77.98% on Zuozhuan data.

Keywords: EvaHan2024, large language model, in-context learning

1. Introduction

Ancient Chinese texts typically consist of only characters without punctuation marks. So researchers in the field of ancient Chinese face the challenge of dealing with large amounts of unpunctuated text. Employing LLMs to do sentence punctuation will save significant manpower and facilitate subsequent research.

The prediction pipeline of our system is shown in figure 1. We take unpunctuated text as input and induce LLM to generate text with proper punctuation by in-context learning(ICL). To construct ICL prompt for each test input separately, we obtain the document category and sentence POS tag sequence by LLM QA, and then select texts with high similarity to the test input from training set. In addition, the generation mechanism of LLMs does not guarantee that the model will give fully standard result, which means the characters may not exactly consistent with the input. For the completeness of our system, we present an efficient and general post-processing scheme based on sequence matching implemented by dynamic programming.

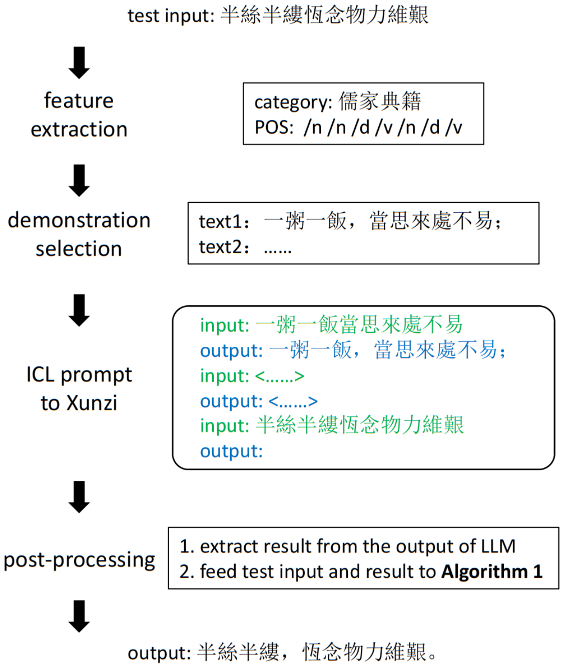

test input: 半絲半縷恆念物力維艱
feature extraction
category: 儒家典籍
POS: /n /n /d /v /n /d /v
demonstration selection
text1：一粥一飯，當思來處不易；
text2：……
ICL prompt to Xunzi
input: 一粥一飯當思來處不易
output: 一粥一飯，當思來處不易；
input: <……>
output: <……>
input: 半絲半縷恆念物力維艱
output:
post-processing
1. extract result from the output of LLM
2. feed test input and result to **Algorithm 1**
output: 半絲半縷，恆念物力維艱。

Figure 1: Prediction pipeline

2. Method

2.1. In-context learning

In-context learning(Dong et al., 2023) is a paradigm that allows LLM to learn tasks given only a few examples, which means, we can concatenate some pairs of input and output from the training set before the test input to help the general LLM perform a specific task better. The choice of examples may affect the performance significantly(Liu et al., 2022). In our system, we choose texts based on similarity. Concretely, we first obtained the category of the documents by LLM QA. Then among the documents of the same type as test input, we compute a similarity score between the POS tagging sequences of test and training text, which are also obtained by LLM QA. Texts with high scores will be used to construct the prompt for Xunzi.

Regarding classification, we predefined some common text categories to provide as options to the LLM. By limiting the scope of the answer, the validity of classification results is basically guaranteed.

LT4HALA 2024@LREC-COLING-2024, pages 261–265
25 May, 2024. © 2024 ELRA Language Resources Association: CC BY-NC 4.0

Regarding POS tagging, due to the long output sequence, illegal tags and unlabeled results are prone to occur. In order to reduce this situation, we randomly selected a labeled result which contains most of the labels from the training set as a reference and add it to the QA prompt. The pos tagging result is in the form of " 半絲/n 半縷/n 恆/d 念/v 物力/n 維/d 艱/v", and we compute similarity score between the label sequences like [n,n,d,v,n,d,v].

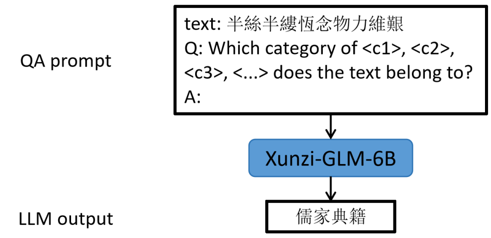

QA prompt
text: 半絲半縷恆念物力維艱
Q: Which category of <c1>, <c2>, <c3>, <...> does the text belong to?
A:
Xunzi-GLM-6B
LLM output
儒家典籍

Figure 2: classification model

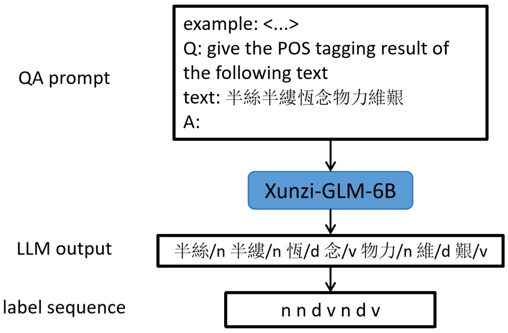

QA prompt
example: <...>
Q: give the POS tagging result of the following text
text: 半絲半縷恆念物力維艱
A:
Xunzi-GLM-6B
LLM output
半絲/n 半縷/n 恆/d 念/v 物力/n 維/d 艱/v
label sequence
n n d v n d v

Figure 3: POS tagging model

For brevity, we use <c1>, <c2>, <c3> in the diagram instead of concrete categories, which are shown in table 1. The randomly selected example in the QA prompt for POS tagging is alse omitted. In addition, since Xunzi is specifically finetuned for ancient Chinese, the actual QA prompts are all completely in Chinese, and the same is true for ICL prompts.

Since we hope the demonstration have strong structural similarity to the test input, at least locally, a score based on the longest common substring(LCS) is adopted. We add the latter term to give a relatively higher score to shorter sequences when the substring length is the same, for less redundant or confusing information, which may be significantly helpful for short text punctuation prediction.

$$l = LCS(POS_{train}, POS_{test})$$
$$s = l + l/LEN(POS_{train})$$

After we have determined the demostrations, the ICL prompt can be constructed as presented in figure1 and fed to the LLM. We remove the punctuation marks in the text to build demonstration inputs and use the original text as output. Then the test input is attached and we expect a well-punctuated output from the LLM, to be specific, Xunzi-GLM-6B in our system.

2.2. Post processing

The results given by the large model are not always completely standardized, such as the mixed use of traditional and simplified Chinese characters, missing and wrong characters, etc. However, we believe that most of the characters in the generated results are still consistent with the input. So we can try to correct the results to ensure that the final results are fully standardized while retaining effective punctuation as much as possible.

As shown in figure 4, we designed a general and efficient scheme. Firstly, the character-by-character alignment results of input (unpunctuated raw text) and output (prediction from LLM) are obtained by the dynamic programming matrix of the longest common subsequence algorithm. Then the aligned parts are kept and the remaining parts are filled with characters in the input and punctuations in the output. The error types can be counted while the final results are obtained. See the pseudocode in Algorithm 1 for details.

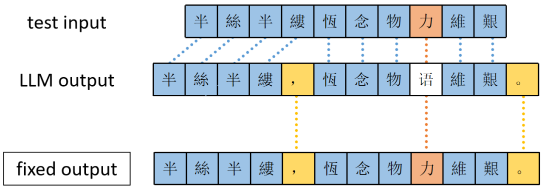

test input
半 絲 半 縷 恆 念 物 力 維 艱
LLM output
半 絲 半 縷 ， 恆 念 物 语 維 艱 。
fixed output
半 絲 半 縷 ， 恆 念 物 力 維 艱 。

Figure 4: Post processing

3. Experiments

Both the classification and POS tagging of the training set documents can be done in advance and the results are saved. In the testing phase, our pipeline is: 1. do classification and POS tagging for the test input; 2. select samples from training documents of same type; 3. construct ICL prompt for the LLM and get the prediction; 4. do post-processing and output the final result.

The experiment was carried out on two NVIDIA RTX A6000 with 48G memory. We have two test dataset. The first dataset consists of several documents of different categories, and we will refer to as test A. The second dataset is Zuozhuan, which is a history book, and we will refer to as test B.

Totally, a prediction of 310,207 tokens is finished in 6600 seconds. With an average of 47 tokens per second, the computational efficiency is acceptable.

Algorithm 1 Post-processing

```
Input: The test input, unpunctuated raw text, $R$; The prediction from llm, $H$; Punctuation mark list, $P$; Alignment result $A = [(i_1, j_1), \ldots, (i_n, j_n)]$
Output: A fixed result $F$ similar to $H$ but consistent with $R$;
$pi = pj = 0$
$F = empty\ string$
for $m = 1ton$ do
    $i, j = A[m]$
    while 1 do
        $ri = R[pi], hj = H[pj]$
        if $pi = i$ then
            if $pj = j$ then
                $F = F + hj,\ pi++,\ pj++$
                break
            else if $hj$ in $P$ then
                $F = F + hj,\ pj++$
            else // redundant character
                $pj++$
            end if
        else
            if $pj = j$ then // missing character
                $F = F + ri,\ pi++$
            else if $hj$ in $P$ then
                $F = F + hj,\ pj++$
            else // wrong character
                $F = F + ri, pi++, pj++$
            end if
        end if
    end while
end for
while $pi < len(R)$ do
    $F = F + R[pi],\ pi++$
end while
while $pj < len(H)$ do
    $hj = H[pj]$
    if $hj$ in $P$ then
        $F = F + hj$
    end if
    $pj++$
end while
    return $F$
```

3.1. Data Information

We predefined 14 categories on the training dataset, and the number of documents and characters for each category are shown in the table 1. The following six categories were involved in the testing stage and star-marked in the table: Confucianism(儒家典籍), Buddhist sutra(佛教經文), Prose(散文雜記), History(歷史), Geography(地理), Agronomy(農學).

In fact, the categories may not cover all documents, and the accuracy of classification results given by LLM is difficult to verify due to the lack of expert knowledge. We believe that when the LLM gives the same classification label to two documents, at least for the model, some features of the pair are consistent, and it is more likely to be useful for our task.

type	docs	tokens
Confucianism*	14	5945471
Novel	25	4111359
Medical	35	3529444
History*	29	3343991
Criticism	36	1657880
Drama	3	1264962
Prose*	26	1030393
Taoist sutra	68	887175
Buddhist sutra*	8	427919
Geography*	4	370990
Poetry	5	220044
Astrology	3	189229
Art of war	6	166254
Agronomy*	4	34117

Table 1: statistical information of training data

The statistical information of the two test datasets is shown in table 2. Test A consists of several short books of different types, while Test B is a single long history book.

	docs	tokens
test A	6	50722
test B	1	199879

Table 2: statistical information of test data

3.2. Results

We get evaluation results of segmentation and punctuation. For segmentation, we treat all punctuation marks as a segment mark to compute the metrics. The baseline model is Xunzi-Qwen-7B-Chat.

	Precision	Recall	F1-Score
SEG	90.53%	66.12%	76.42%
PUNC	73.52%	52.22%	61.06%

Table 3: Test A, baseline(Xunzi-Qwen-7B-Chat)

	Precision	Recall	F1-Score
SEG	95.28%	87.17%	91.04%
PUNC	79.25%	72.09%	75.50%

Table 4: Test B, baseline(Xunzi-Qwen-7B-Chat)

Our system adopted Xunzi-GLM-6B as base model since it tends to generate relatively standard results. To verify the effectiveness of our strategy for selecting demonstrations, we conducted the following two sets of experiments for comparison. To ensure the standardability of the prediction, the outputs were all post-processed.

For experiment 1, when building the ICL prompt, we used BM2.5 to retrieve highly related text among documents of the same type. BM2.5 is a statistical method based on word frequency, which means it may have better efficiency but will ignore the sequence information of texts. From the results, the model works well with ICL and the simply constructed prompts on Test A, while has no positive effect on Test B, which seems to be easier from the baseline.

	Precision	Recall	F1-Score
SEG	91.54%	73.54%	81.56%
PUNC	74.52%	58.44%	65.51%

Table 5: Exp 1, Test A, Xunzi-GLM-6B, with ICL prompt build by classification (LLM QA) and retrieval (BM2.5)

	Precision	Recall	F1-Score
SEG	95.53%	86.56%	90.82%
PUNC	79.72%	72.18%	75.76%

Table 6: Exp 1, Test B, Xunzi-GLM-6B, with ICL prompt built by classification (LLM QA) and retrieval(BM2.5)

For experiment 2, We used exactly the same system as described in the former section. We further exploited the ability of the LLM to obtain POS tag sequences and designed a similarity metric for demonstration selection. From the results, we can see that with the higher level feature, the performance of the system is improved on both test sets.

	Precision	Recall	F1-Score
SEG	89.65%	79.49%	84.27%
PUNC	72.87%	63.25%	67.72%

Table 7: Test A, Exp 2, Xunzi-GLM-6B, with ICL prompt built by classification (LLM QA), POS tagging (LLM QA) and similarity

	Precision	Recall	F1-Score
SEG	95.38%	89.68%	92.44%
PUNC	80.44%	75.67%	77.98%

Table 8: Test B, Exp 2, Xunzi-GLM-6B, with ICL prompt built by classification (LLM QA), POS tagging (LLM QA) and similarity

Finally, we add a note on the role of post-processing in the system. We processed the output line by line. In Exp 2, Test B contains 3319 lines, of which 2828 lines were standard, and the remaining 491 lines were output after post-processing, accounting for about 15%. The proportion was even higher for Test A, with 293 standard lines among a total of 401 lines, 27% of the output were post-processed.

4. Discussion

For the detailed results, the performance of the system on different types of text does vary significantly. In test A, we achieved an F1 score of 61.61% for the Buddhist sutra(佛教经文) type, which is well below average(67.72%) and Test B(77.98%). But this type also shows the most significant improvement over baseline(55.73%). This indicates that the large language model itself is less capable of handling this type of text, which is related to the obscure language and unusual expression of Buddhist texts.

In general, we adopted ICL framework in our system and selected texts that are similar to the test input as demonstrations. We first narrow the range with categories and then perform fine-grained matching by POS sequences. The combination of content and structure features achieved good results. However, ICL demonstrations can also be selected based on other criterias, such as annotation difficulty(Drozdov et al., 2023) or the proportion of different punctuation marks in the text(Levy et al., 2023). Moreover, the order of examples can also affect the generation results(Lu et al., 2022).

Our system made use of the large language model's own knowledge and general ability to compensate for the lack of external domain knowledge. We used the features obtained by LLM QA to construct prompts for the same model, which is kind of accommodation to the model. In experiments with small models, we worry about error propagation between two stages, but for larger models, this potential consistency may tend to have a positive impact as the model becomes more powerful.

5. References

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Andrew Drozdov, Honglei Zhuang, Zhuyun Dai, Zhen Qin, Razieh Rahimi, Xuanhui Wang, Dana Alon, Mohit Iyyer, Andrew McCallum, Donald

Metzler, and Kai Hui. 2023. PaRaDe: Passage ranking using demonstrations with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14242–14252, Singapore. Association for Computational Linguistics.

Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1401–1422, Toronto, Canada. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Author Index