Corpus Services: a Framework to Curate XML Corpus Data

Aleksandr Riaposov, Elena Lazarenko
Universität Hamburg
Hamburg, Germany
{aleksandr.riaposov, elena.lazarenko}@uni-hamburg.de

Abstract

This paper provides a comprehensive description of the Corpus Services framework—a collection of Java validation tools for language corpora compiled in XML-based data formats, in particular those used by EXMARaLDA corpus software. Having successfully found application in several research projects, the core functionality of the framework is currently integrated in the automated curation and publication workflows for EXMARaLDA-driven corpora of Northern Eurasian languages, as developed by the long-term project INEL. Preliminary stages of development and examples of practical use cases are covered, a structured explanation of the framework's current functionality and operational mechanisms is provided. Furthermore, the utilization of Corpus Services is extensively illustrated within the context of INEL workflows.

Keywords: language corpora, data curation, quality assurance

1. Introduction

The long-term project INEL (Grammatical Descriptions, Corpora, and Language Technology for Indigenous Northern Eurasian Languages)[1] commenced in 2016. It conducts an extensive empirical analysis of linguistic data sourced from endangered (or even extinct) language varieties of Northern Eurasia. The language families that the project deals with at the moment vary from Uralic to Turkic and Tungusic. There are several corpora already published for Samoyedic (< Uralic) languages, namely the INEL Kamas Corpus and the INEL Selkup Corpus (Gusev et al., 2023; Brykina et al., 2021) with more corpora on other related languages currently underway. The former comprises data mainly from the last Kamas speaker Klavdiya Plotnikova alongside with texts from earlier period collected by Kai Donner and the latter is composed of texts from the archive of Angelina Ivanovna Kuzmina who collected Selkup materials in 1962-1977. The Turkic language family is presented in the project by the INEL Dolgan Corpus that is composed of data from various time periods (1970-2000s, 2007-2010, 2017) (Däbritz et al., 2022). Tungisic languages are represented by the INEL Evenki Corpus that is as well complied from several text sources and time periods (Däbritz and Gusev, 2021). In the current project phase, corpora of three more Samoyedic languages are under development—Enents, Nenets, and Nganasan—and a new extended version of the INEL Evenki Corpus will be delivered. The key focus during the corpus creation is to produce deeply annotated XML-based language corpora and accompanying resources and to make them sustainably available. The digital workflows adopted by the project are thoughtfully tailored to accommodate the diversity and multimodality of the data types incoming: corpora of minority languages contain a lot of manually collected, transcribed, and annotated sources that can exhibit a significant level of heterogeneity and require meticulous attention to detail in order to make the resulting data usable by a broader audience. Unfortunately, no out-of-the-box solution from the world of language documentation and corpus linguistics could effectively be integrated into the data curation process. While the EXMARaLDA suite (Schmidt and Wörner, 2014) offers a set of validation tools for the corpus files, these tools do not cover all the situations applicable for the multimodal data INEL is developing. Therefore, a significant part of automatic data curation spanning from the first steps of data preprocessing to consistency checks and eventual publication has been covered by the Corpus Services framework (Arkhangelskiy et al., 2020). It is a Java collection of data validators for XML-based formats, primarily those used by the EXMARaLDA suite. The original set of Corpus Services validators has been developed under the auspices of *Hamburg Centre for Language Corpora (HZSK)*[2] and has received contributions from INEL, the BMBF-funded CLARIN-D project[3], the project WO 1886/1-2 within the DFG LIS program, and the BMBF-funded project QUEST[4]. Within the INEL project the framework has been adapted to cover project-specific tasks and currently includes

[1] https://www.slm.uni-hamburg.de/inel.html

[2] https://www.slm.uni-hamburg.de/hzsk.html

[3] https://www.clarin-d.net/en/clarin-d/project-summary

[4] https://www.slm.uni-hamburg.de/ifuu/forschung/forschungsprojekte/quest.html

LREC-COLING 2024, pages 4030–4035

data validators that run nightly in fixing or reporting modes, which are meant to alleviate the linguists' work. Parts of Corpus Services were integrated within the Git client LAMA (Riaposov et al., 2022) conceptualized by the HZSK, and developed to become a part of the INEL digital workflows. The Corpus Services framework[5] is open-source and licensed under the MIT License—henceforth can be adapted to validate other EMARaLda-based or even XML-based linguistic corpora.

2. Preliminary work

2.1. HZSK period

Hamburg Centre for Language Corpora (HZSK) has been working on the problem of elevating quality management and quality assurance for linguistic resources for over a decade. The reason behind it was that curation of language data, especially in case of lesser-described languages, had been heavily relying on manual work and could not be completely automated due to the data structure, thus preparation of a corpus for dissemination could become a long, drawn-out process with heavy workload (Ferger and Hedeland, 2020). In order to address these challenges HZSK made a decision to incorporate parts of software development workflow into the language corpus development workflow. This would primarily mean introduction of version control and continuous integration throughout the corpus preparation (Ferger and Hedeland, 2020). The latter was backed up by the conceptualization and building of a Java quality control software framework—so-called *HZSK Corpus Services* or later just *Corpus Services*—that at the early development stages already provided a set of semi-automatic quality control tools that would address data curation issues. It was based on the code of EXMARaLDA (see 2.2) but would enhance it with further features that were not directly available within the EXMARaLDA suite and would make data consistent on the fly, or report problems that could not be handled automatically and needed manual corrections. Moreover, the quality control was additionally backed up by versioning and issue tracking (Ferger and Hedeland, 2020; Hedeland, 2020) Since then, Corpus Services has been undergoing continuous development and expansion to satisfy the needs of the projects that are using the framework (see 3.1).

2.2. EXMARaLDA

The structure of the corpora developed under the roof of HZSK and currently within the INEL project is based upon EXMARaLDA, an open-source software suite with its own set of XML-based data formats (Ferger and Hedeland, 2020; Schmidt and Wörner, 2014). Generally speaking, the heart of an EXMARaLDA corpus is a corpus metadata file (comafile) that has its own XML-based format (.coma) and is managed with the help of *EXMARaLDA Corpus-Manager* (*Coma*). A comafile contains all the relevant information about the corpus itself, speakers, communications, and transcriptions. Transcriptions linked to the comafile are stored in the corpus as separate files of two XML-based formats: EXMARaLDA basic transcription (.exb) and EXMARaLDA segmented transcription (.exs). As a rule an .exb file has a main transcription tier and further optional tiers of other types. It is the file format a linguist works with firsthand using *EXMARaLDA Partitur-Editor*. The segmented transcription files carry information about the building blocks of a transcription, e.g. single utterances containing words. The segmentation of .exb files into .exs ones can be performed automatically with the help of several algorithms[6]. INEL uses a customized version of the HIAT algorithm[7]. The .exs files thus generated are needed for *EXAKT—EXMARaLDA Analysis and Concordance tool*[8]; besides that, XML files of the ISO/TEI format are produced from .exs. ISO/TEI files, in turn, are ingested into the Tsakorpus platform[9] where the INEL corpora can be browsed online (Arkhangelskiy et al., 2019).

3. Corpus Services

3.1. History and Use Cases

After its debut in HZSK, contributions to the framework have been made not only by HZSK staff, but also by several research projects at and outside of the University of Hamburg. The Corpus Services framework came to be utilized mainly in two projects, and the use cases eventually diverged to an extent. The project QUEST (see Arkhangelskiy et al. 2020, Wamprechtshammer et al. 2022) was working towards Corpus Services universal usability by the linguistic community: the original Java application was wrapped in a web GUI that walked a user through the process of data curation. Specifically, that meant that the user either compiled a questionnaire about their dataset and allowed the system to automatically generate curation settings based on the answers, or chose the settings manually (Arkhangelskiy et al., 2020). One of the QUEST-oriented development and use cases is the RefCo (Reference Corpora) checker that extends the processed formats primarily with the ELAN one and can be adapted to further data

[5] https://gitlab.rrz.uni-hamburg.de/corpus-services/inel-corpus-services

[6] See How to Use Segmentation

[7] See Overview of HIAT transcription convention

[8] See Understanding the basics of EXMARaLDA

[9] https://github.com/timarkh/tsakorpus

formats (Lange and Aznar, 2022).
The long-term project INEL has taken a slightly different approach in utilizing Corpus Services. The automatic data curation process, version control, and issue tracking have become an integral part of the corpus preparation and publication workflows. Due to the fact that the languages that the project deals with are extremely low-resources, the INEL workflows require frequent input from linguists and partially rely on manual or semi-automatic processes during the initial stages of data preparation (e.g. stages "Raw data" and "Working data 1: Grammatical analysis" of 1 and 2 often require manual transcription of files or manual grammatical glossing of the obtained data prior to the automatic curation), they benifit a lot from the automatic corpus curation at the later stages. This results in improved data consistency throughout different corpora, less time required to deliver a final product, and no data loss.

3.2. INEL Corpus Services

While the earlier version of Corpus Services helped to lay down the basic principles used in the workflow, in practice it ran into performance issues: first, the Java VM would consume unreasonable amounts of heap memory, and second, because of inefficiencies in the code base, the validation process would take a lot of time. As the amount of data being checked by the framework grew, the constraints stipulated above have proven the continuous usage of the HZSK version to be unfeasible. In order to address this, we developed a new code base that keeps the core functionality intact while boasting significantly better performance. The data on comparative performance for both versions are reflected in Table 1 and Table 2.

Corpus Services version	Runtime (s)	JVM heap memory peak (Mb)	JVM average heap memory (Mb)
CS-INEL	168	415.74	239.23
CS-old	1508	3859.36	3008.17

Table 1: Benchmark tests for INEL Selkup Corpus 2.0 (Corpus size: 81498 words, 352 EXBs) (Brykina et al., 2021)

At the moment INEL Corpus Services has successfully replaced the HZSK version during the curation process of all the corpora under development; a new version of INEL Kamas Corpus has been published, using INEL Corpus Services both during curation and publication stages.

How the framework operates

Corpus Services version	Runtime (s)	JVM heap memory peak (Mb)	JVM average heap memory (Mb)
CS-INEL	150	197.74	124.34
CS-old	1958	4248.82	3520.61

Table 2: Benchmark tests for INEL Dolgan Corpus 2.0 (Corpus size: 97757 words, 136 EXBs) (Däbritz et al., 2022)

- The main class CorpusServices reads parameters from the command line: path to the directory, where the files to be checked are located; the list of functions to be performed; (optional) switch to allow automatic corrections of the files; (optional) path to the output file; (optional) parameters to be passed down to the functions.
- The class CorpusFunctions recursively goes over the file tree under the input directory looking for files with relevant extensions, and, whenever it encounters such a file, reads it as a DOM Document object, which is then passed to each applicable function that was called in the command. The process repeats until the list of commands is exhausted; then the framework moves on to the next file.
- The classes in .validation and .utilities packages—also referred to as functions—parse the Document object, perform actions specified in the function body, and return an updated Document to be processed by the next function in the queue. Whenever an error is found in the data, a function will call the ReportItem class.
- The class ReportItem processes errors and throws each one back to the main class, which stores them in a temporary JSON-like structure while the program continues to execute.
- Finally, CorpusServices writes out the list of errors in a JSON or HTML file.

What makes the the Corpus Services framework to stand out is its language-independency: due to the fact that it concentrates on finding errors in the corpus structure and improving data consistency throughout it, it can be used nearly as-is to curate corpora of other languages as long as they are built with the EXMARaLDA software.

Available functionality There is a number of trade-offs that have to be made when designing a framework such as Corpus Services. While it may seem enticing to employ an all-encompassing approach aimed at development of a framework that

would be pliable enough to curate sundry XML-based corpora out of the box, no matter the particular XML specification used, the amount of annotations per sentence provided, the way the corpus metadata are presented, or the kind of corpus in general, such maximalist ambitions prove to be deeply impractical given that the real-life framework application scenarios, at least for the time being, target a rather narrow range of corpora. With that in mind, the design principle of Corpus Services was to prioritize the needs of people already using it while leaving the door open for easy extension of the available functionality, should more users get interested in contributing to the framework. The following is a list of possibilities Corpus Services already offers to its users:

- Run an XSL transformation given a stylesheet, or an external XQuery script;
- Conversion utilities: split a flextext file containing multiple texts that is generated in FLeX into separate flextext files, one for each text, to be then imported into EXMARaLDA; transform a flextext file to the EXMARaLDA Basic Transcription format (.exb);
- Curation utilities: available only for the XML specifications supported by the EXMARaLDA software suite, i.e., EXMARaLDA Basic Transcription data (.exb), EXMARaLDA Segmented Transcription data (.exs), and the metadata (.coma). The utilities may be grouped into (i) garbage removal that cleans up unnecessary, empty, and problem-inducing symbols and tags from the data, (ii) file coverage checks ensuring that every file in the corpus directory is mentioned in the metadata, and, vice versa, that every filepath referred to in the metadata is relative and resolves to an existing file, (iii) formatting checks that bring formalized chunks of data such as tier names, speaker abbreviations, format tables, or reference tier IDs to a consistent and uniform appearance, (iv) structure checks looking for annotation mismatches, inconsistencies on the timeline, and utterance end symbols that have to appear at the end of very sentence and only there;
- Automatic segmentation, i.e., conversion from Basic Transcription data to Segmented Transcription data with customizable parameters for the segmentation algorithm and the finite state machine used in the process;
- Glossing utilities: as EXMARaLDA Partitur-Editor does not provide a way to change glossing across the entire corpus, Corpus Services attained the much-needed ability to rename, merge and delete specified glosses with the possibility to check for context on another tier if necessary.

4. INEL Corpus Services workflow

Corpus Services comes into play at several different places within the INEL workflow (see 1 and 2). Firstly, it may be used at the stage where the preliminary data are being converted to the EXMARaLDA Basic Transcription format, especially when the built-in import functionality of EXMARaLDA proves to be insufficiently convenient (e.g., when importing multiple files at the same time), or when the file to be imported requires certain preprocessing (relevant classes: Flextext2EXBMultiConverter, FlextextSplit).
Secondly, during the lengthy annotation process each INEL corpus is stored in a respective Git repository. While version control provided by Git allows multiple people to work on the corpus data at the same time, with the added benefit of each change being tracked, Git also puts extra requirements on the file formatting: in order to prevent the system to be inundated with errors caused by differences in whitespace and newlines, the files are uniformly pretty-printed by Corpus Services after each commit a project member makes. To facilitate this, and the work with Git in general, an in-house Git client LAMA was developed (Riaposov et al., 2022). Besides acting as a way to interface with Git, LAMA is used to manually run selected Corpus Services functions (relevant classes: PrettyPrintData, ExbNormalizeTimeline, ExbReplaceGlosses).
Thirdly, each night Corpus Services runs validation checks on the data, provided that there were changes made in the repository during the previous day. Some checks are fully automated and fix the discrepancies in the data as Corpus Services finds them. In cases where automatic correction is not feasible, Corpus Services tracks errors discovered in the data, and composes a human-readable report in the form of an HTML page (or a machine-readable JSON file) as the output. The report contains information about the error type and its severity (error versus warning), the name of the file where a problem was found, and, where applicable, points at the relevant sentence ID and/or the timeline ID, so that a linguist can easily get to the root of the problem. The errors are supposed to be eliminated from the data before a corpus would be deemed ready for publication (relevant classes: everything included in the .validation package).
Fourthly, Corpus Services aids in polishing the data during the publication process itself. The errors found in the corpus are eliminated to the maximum possible extent, and then the data are normalized, packed in repository-ready archives, or

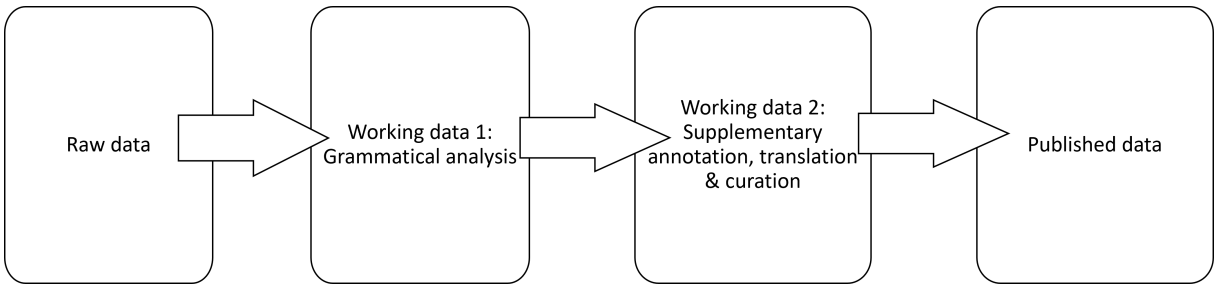

Figure 1: INEL workflow: data types

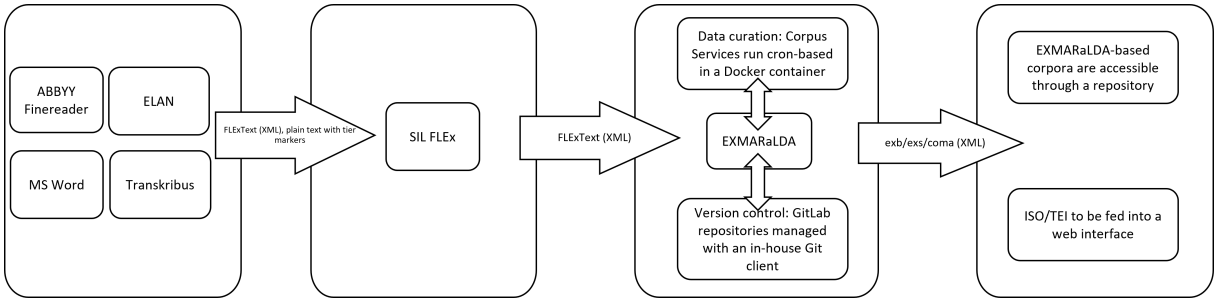

Figure 2: INEL workflow: data formats

converted to be further ingested into the Tsakorpus platform and be availbale not only in downloadable archives but also online directly from the browser. A lightweight option for accessing transcriptions without a search option from the web can be provided by generating simple visualizations of the transcriptions and the Comafile.

5. Conclusion and future work

In the paper we presented Corpus Services, a framework for corpus data curation and validation, briefly outlining its history, scope, and use cases. While Corpus Services successfully tackles on the tasks arising in the INEL workflow, the general unavailability of a tool that would be handling the quality of corpus data in many different contexts remains an issue. With that in mind, we argue that a bottom-up approach that above all accounts for the peculiarities of a specific kind of corpus, while simply being more practical given limited resources, also allows to re-use and adapt the framework to other contexts with greater ease. The vector of future Corpus Services development depends on its wider uptake by the community.

6. Acknowledgements

The long-term project INEL is funded by the German Federal Government and Federal States in the Academies' Programme, with funding from the Federal Ministry of Education and Research and the Free and Hanseatic City of Hamburg. The Academies' Program is coordinated by the Union of the German Academies of Sciences and Humanities.

7. Bibliographical references

Timofey Arkhangelskiy, Anne Ferger, and Hanna Hedeland. 2019. Uralic Multimedia Corpora: ISO/TEI Corpus Data in the Project INEL. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages, January 7-January 8, 2019, Tartu, Estonia*, pages 115–124. Association for Computational Linguistics.

Timofey Arkhangelskiy, Hanna Hedeland, and Aleksandr Riaposov. 2020. Evaluating and Assuring Research Data Quality for Audiovisual Annotated Language Data. In *CLARIN Annual Conference*, pages 1–7.

Anne Ferger and Hanna Hedeland. 2020. Towards Continuous Quality Control for Spoken Language Corpora. *Int. J. Digit. Curation*, 15(1):1–13.

Hanna Hedeland. 2020. Providing Digital Infrastructure for Audio-Visual Linguistic Research Data with Diverse Usage Scenarios: Lessons Learnt. *Publications*, 8(2).

Herbert Lange and Jocelyn Aznar. 2022. RefCo and its Checker: Improving Language Documentation Corpora's Reusability Through a Semi-Automatic Review Process. In *Proceedings of the Thirteenth Language Resources*

and Evaluation Conference, pages 2721–2729, Marseille, France. European Language Resources Association.

Aleksandr Riaposov, Elena Lazarenko, and Timm Lehmberg. 2022. Bringing Together Version Control and Quality Assurance of Language Data with LAMA. In *Proceedings of the Workshop on Resources and Technologies for Indigenous, Endangered and Lesser-resourced Languages in Eurasia within the 13th Language Resources and Evaluation Conference*, pages 36–41, Marseille, France. European Language Resources Association.

Thomas Schmidt and Kai Wörner. 2014. EXMARaLDA. In Jacques Durand, Irike Gut, and Gjert Kristoffersen, editors, *Handbook on Corpus Phonology*, pages 402–419. Oxford University Press.

Anna Wamprechtshammer, Elena Arestau, Jocelyn Aznar, Hanna Hedeland, Amy Isard, Ilya Khait, Herbert Lange, Nicole Majka, and Felix Rau. 2022. QUEST: Guidelines and Specifications for the Assessment of Audiovisual, Annotated Language Data. *Working Papers in Corpus Linguistics and Digital Technologies: Analyses and Methodology*, 8.

8. Language Resources

Brykina, Maria and Orlova, Svetlana and Wagner-Nagy, Beáta. 2021. *INEL Selkup Corpus (Version 2.0)*. [link].

Däbritz, Chris Lasse and Gusev, Valentin. 2021. *INEL Evenki Corpus*. [link].

Däbritz, Chris Lasse and Kudryakova, Nina and Stapert, Eugénie. 2022. *INEL Dolgan Corpus (Version 2.0)*. [link].

Gusev, Valentin and Klooster, Tiina and Wagner-Nagy, Beáta. 2023. *INEL Kamas Corpus (Version 2.0)*. [link].