LREC 2018 Workshop

4REAL 2018

Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language

PROCEEDINGS

Edited by

António Branco, Nicoletta Calzolari and Khalid Choukri

ISBN: 979-10-95546-21-4
EAN: 9791095546214

12 May 2018

Proceedings of the LREC 2018 Workshop

"4REAL 2018 –Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language"

12 May 2018 – Miyazaki, Japan

Edited by António Branco, Nicoletta Calzolari and Khalid Choukri

http://4real2018.di.fc.ul.pt/

Organising Committee

António Branco	University of Lisbon
Nicoletta Calzolari	ILC-CNR/ELRA
Khalid Choukri	ELDA

Programme Committee

Aljoscha Burchardt	DFKI
António Branco	University of Lisbon
German Rigau	Universit of the Basque Counry
Gertjan van Noord	University of Groningen
Joseph Mariani	CNRS/LIMSI
Kevin Cohen	University of Colorado
Khalid Choukri	ELRA
Maria Gavrilidou	ILSP
Marko Grobelnik	Jozef Stefan Institute
Marko Tadić	University of Zagreb
Nancy Ide	Vassar College
Nicoletta Calzolari	ILC-CNR/ELRA
Patrick Paroubek	CNRS/LIMSI
Piek Vossen	VU University Amsterdam
Senja Pollak	Jozef Stefan Institute
Simon Krek	Jozef Stefan Institute
Stelios Piperidis	ILSP
Thierry Declerck	DFKI
Yohei Murakami	Language Grid Japan

Introduction

Reproduction and replication of research results are at the heart of the validation of scientific knowledge and of the scientific endeavor. Reproduction of results entails arriving at the same overall conclusions that is, to appropriately validate a set of results, scientists should strive to reproduce the same answer to a given research question by different means, e.g. by reimplementing an algorithm or evaluating it on a new dataset, etc. Replication has a more limited aim, typically involving running the exact same solution or approach under the same conditions in order to arrive at the same output result.

The immediate motivation for the increased interest on reproducibility and replicability is to be found in a number of factors, including the realization that for some published results, their replication is not being obtained (e.g. Prinz *et al.*, 2011; Belgley and Ellis, 2012); that there may be problems with the commonly accepted reviewing procedures, where deliberately falsified submissions, with fabricated errors and fake authors, get accepted even in respectable journals (e.g. Bohannon, 2013); that the expectation of researchers *vis a vis* misconduct, as revealed in inquiries to scientists on questionable practices, scores higher than one might expect or would be ready to accept (e.g. Fanelli, 2009); among several others.

This workshop sought to contribute to the discussion and the advancement on a topic that has been given insufficient attention in the research area of language processing tools and resources and that has been an important topic emerging in other scientific areas, continuing the objectives of the first edition of the 4REAL workshop, at LREC 2016. We invited the submission of articles that present cases, either with positive or negative results, of actual replication or reproduction exercises of previous published results in our area.

The present volume gathers the papers that were entered as anonymous submissions and that received sufficiently positive evaluation by three reviewers of the workshop's program committee. We hope that it may foster the replication and replicability of research results in the field of science and technology of language.

12 May 2018

António Branco, Nicoletta Calzolari and Khalid Choukri

References

Begley and Ellis, 2012, "Drug development: Raise standards for preclinical cancer research", *Nature*.

Bohannon, John, 2013, "Who's Afraid of Peer Review?", *Science*.

Fanelli, 2009, "How Many Scientists Fabricate and Falsify Research? A Systematic review and meta-analysis of survey data". PL OS ONE.

Prinz, *et al.*, 2011, "Believe it or not: how much can we rely on published data on potential drug targets?", *Nature Reviews Drug Discovery*.

Programme

9:00 – 9:10 – Welcome address

9:10 – 10:40 – Session 1

Andraž Repar, Matej Martinc and Senja Pollak,
Machine Learning Approach to Bilingual Terminology Alignment: Reimplementation and adaptation

João Silva, João António Rodrigues, Vladislav Maraev, Chakaveh Saedi and António Branco,
A 20% Jump in Duplicate Question Detection Accuracy? Replicating IBM team's experiment and finding problems in its data preparation

Filip Klubička and Raquel Fernández,
Examining a Hate Speech Corpus for Hate Speech Detection and Popularity Prediction

10:40 – 11:10 – Coffee break

11:10 – 12:10 – Session 2

Steve Cassidy and Dominique Estival,
Annotation Contributions: Sharing derived research data

Simone Fuscone, Benoit Favre and Laurent Prévot,
Replicating Speech Rate Convergence Experiments on the Switchboard Corpus

12:10 – 13:00 – Invited talk

Mark Liberman,
Validation of Results in Linguistic Science and Technology: Terminology, problems, and solutions

13:00 – Farewell

Table of Contents

Machine Learning Approach to Bilingual Terminology Alignment: Reimplementation and Adaptation

Andraž Repar[1,3], **Matej Martinc**[1,2], **Senja Pollak**[2]
[1] Jožef Stefan Postgraduate School, Ljubljana, Slovenia
[2] Jožef Stefan Institute, Ljubljana, Slovenia
[3] Iolar d.o.o., Ljubljana, Slovenia
andraz.repar@iolar.com,{matej.martinc, senja.pollak}@ijs.si

Abstract

In this paper, we reproduce some of the experiments related to bilingual terminology alignment described by Aker et al. (2013). They treat bilingual term alignment as a binary classification problem and train a SVM classifier on various dictionary and cognate-based features. Despite closely following the original paper with only minor deviations - in areas where the original description is not clear enough - we obtained significantly worse results than the authors of the original paper. In the second part of the paper, we try to analyze the reasons for the discrepancy and offer some methods to improve the results. After improvements we manage to achieve a precision of almost 91% and recall of almost 52% which is closer to the results published in the original paper. Finally, we also performed manual evaluation where we achieved results similar to the original paper. To help with any future reimplementation efforts of our experiments, we also publish our code online.

Keywords: term alignment, machine learning, SVM, cognates, word alignment dictionary

1. Introduction

As part of a larger body of work related to bilingual terminology extraction for the needs of the translation industry, we were interested in implementing a machine learning approach to bilingual terminology alignment. The primary purpose of bilingual terminology alignment is to build a term bank - i.e. a list of terms in one language along with their equivalents in the other language. With regard to the input text, we can distinguish between alignment on the basis of a parallel corpus and alignment on the basis of a comparable corpus. For the translation industry, bilingual terminology extraction from parallel corpora is extremely relevant due to the large amounts of sentence-aligned parallel corpora available in the form of translation memories (in the TMX file format). Foo (2012) makes a distinction between two basic approaches: *Extract-align* where we first extract monolingual candidate terms from both sides of the corpus and then align the terms, such as in Vintar (2010), and *align-extract* where we first align single and multi-word units in parallel sentences and then extract the relevant terminology from a list of candidate term pairs, such as in Macken et al. (2013).

However, considerable efforts have also been invested into researching terminology alignment from comparable corpora Daille and Morin (2005) state that there are multiple reasons why one would opt to extract terminology from comparable and not parallel corpora with the most important being that it is often difficult to obtain parallel corpora not involving English. One of the approaches to term alignment on the basis of comparable corpora involves cognates - words that look similar in different languages (e.g. "democracy" in English and "demokracija" in Slovenian), for example Mann and Yarowsky (2001) describe a method that uses cognates to generate bilingual lexicons between languages from different language families.

In this paper, we aim to reproduce the experiments from the paper "Extracting bilingual terminologies from comparable corpora" by Aker et al. (2013) who propose an original approach to bilingual term alignment utilizing machine learning techniques. They treat aligning terms in two languages as a binary classification problem and employ an SVM binary classifier (Joachims, 2002) and training data terms taken from the EUROVOC thesaurus (Steinberger et al., 2002). They construct two types of features: dictionary-based (using word alignment dictionaries created with Giza++ (Och and Ney, 2000; Och and Ney, 2003) and cognate-based (effectively utilizing the similarity of terms across languages).

Despite the problem of bilingual term alignment lending itself well to the binary classification task, there have only been relatively few approaches utilizing machine learning. For example, similar to Aker et al. (2013), Baldwin and Tanaka (2004) generate corpus-based, dictionary-based and translation-based features and train a SVM classifier which returns a continuous value between -1 and 1 which in turn is then used to rank the translation candidates. Note that they only focus on multi-word noun phrases (noun + noun). A similar approach, again focusing on noun phrases, is also described by Cao and Li (2002). Finally, Nassirudin and Purwarianti (2015) also reimplement the approach by Aker et al. (2013) for the Indonesian-Japanese language and further expand it with statistical features (i.e. context heterogeneity similarity). In the best scenario, their accuracy, precision and recall all exceed 90% but the results are not directly comparable since Nassirudin and Purwarianti (2015) use 10-fold cross-validation while Aker et al. (2013) use a held-out test set.

This paper is organized as follows: Section 1 contains the introduction, Section 2 describes the approach by Aker et al. (2013), Section 3 contains our reimplementation efforts, Section 4 describes the approach to improve the reimplementation results, Section 5 contains the results of manual

evaluation and Section 6 contains the conclusions. We also publish our code online for enabling future replicability [1].

2. Description of the original approach

The original approach designed by Aker et al. (2013) was developed to align terminology from comparable (or parallel) corpora using machine-learning techniques. They use terms from the EUROVOC thesaurus and train an SVM binary classifier (Joachims, 2002) (with a linear kernel and the trade-off between training error and margin parameter c = 10). The task of bilingual alignment is treated as binary classification - each term from the source language *S* is paired with with each term from the target language *T*. They then extract features (dictionary and cognate-based) to be used by the classifier. They run their experiments on the 21 official EU languages covered by EUROVOC with English always being the source language (20 language pairs altogether). They evaluate the performance on a held-out term pair list from EUROVOC using recall, precision and F-measure for all 20 languages. Next, they propose an experimental setting for a simulation of a real-world scenario where they collect English-German comparable corpora of two domains (IT, automotive) from Wikipedia, perform monolingual term extraction (based on Pinnis et al. (2012)), followed by the bilingual alignment procedure described above and manually evaluate the results (using two evaluators). They report excellent performance on the held-out term list with many language pairs reaching 100% precision and the lowest recall being 65%. For Slovene, the target language of our interest, the results were 100% precision and 66% recall. The results of the manual evaluation phase were also good, with two evaluators agreeing that at least 81% of the extracted term pairs in the IT domain and at least 60% of the extracted term pairs in the automotive domain can be considered exact translations.

2.1. Features

Aker et al. (2013) use two types of features that express correspondences between the words (composing a term) in the target and source language (for a detailed description see Table 1:

- 7 dictionary-based (using Giza++) features[2] which take advantage of dictionaries created from large parallel corpora of which 6 are direction-dependent (source-to-target or target-to-source) and 1 direction-independent - resulting in altogether 13 features, and
- 5 cognate-based (on the basis of Gaizauskas et al. (2012)) which utilize string-based word similarity between languages.

To capture words with morphological differences, they do not perform direct string matching but utilize Levenshtein Distance. Two words were considered equal if the Levenshtein Distance (Levenshtein, 1966) was equal or higher than 0.95.
Additional features are also constructed by:

- Using language pair specific transliteration rules to create additional cognate-based features. The purpose of this task was to try to match the cognate terms while taking into account the differences in writing systems between two languages. Transliteration rules were created for both directions (source-to-target and target-to-source) separately and cognate-based features were constructed for both directions - resulting in additional 10 cognate-based features with transliteration rules.
- Combining the dictionary and cognate-based features in a set of combined features where the term pair alignment is correct if either the dictionary or the cognate-based method returns a positive result. This process resulted in additional 10 combined features[3].

At the end of the feature construction phase, there were 38 features: 13 dictionary-based, 5-cognate-based, 10 cognate-based features with transliteration rules and 10 combined features.

2.2. Data sources and experiments

Using Giza++, Aker et al. (2013) create source-to-target and target-to-source word alignment dictionaries based on the DGT translation memory (Steinberger et al., 2002). The resulting dictionary entries consist of the source word *s*, its translation *t* and the number indicating the probability that *t* is an actual translation of *s*. To improve the performance of the dictionary-based features, the following entries were removed from the dictionaries:

- entries where probability is lower then 0.05
- entries where the source word was less than 4 characters and the target word more than 5 characters long and vice versa.

The next step is the creation of term pairs from the EUROVOC thesaurus, which at the time consisted of 6,797 terms. Each non-English language was paired with English. The test set consisted of 600 positive (correct) term pairs—taken randomly out of the total 6,797 EUROVOC term pairs—and around 1.3 million negative pairs which were created by pairing a source term with 200 distinct random terms. Aker et al. (2013) argue that this was done to simulate real-world conditions where the classifier would be faced with a larger number of negative pairs and a comparably small number of positive ones. The 600 positive term pairs were further divided into 200 pairs where both (i.e. source and target) terms were single words, 200 pairs with a single word only on one side and 200 pairs with multiple-word terms on both sides. The remaining positive term pairs (approximately 6,200) were used as training data along with additional 6,200 negative pairs. These were constructed by taking the source side terms and pairing

[1] http://source.ijs.si/mmartinc/4real2018

[2] For languages like German, with extensive usage of compounding, additional rules are applied.

[3] For combined features, a word is considered as covered if it can be found in the corresponding set of Giza++ translations or if one of the cognate-based measures (Longest Common Subsequence, Longest Common Substring, Levensthein Distance, Needleman-Wunsch Distance, Dice) is 0.70 or higher (set experimentally by Aker et al. (2013))

each source term with one target term (other than the correct one). Using this approach, Aker et al. (2013) achieve excellent results with results for Slovenian reaching 100% precision and 66% recall.

3. Reimplementation of the approach

As part of a larger body of work on bilingual terminology extraction, we find machine learning approaches interesting because they allow continuous improvement of the output either by fine-tuning or customizing the training set to the output requirements. For this purpose, the approach by Aker et al. (2013) represents a fine starting point for machine-learning-based bilingual term alignment.

The first step in our approach was to reimplement the algorithm described by Aker et al. (2013). The initial premise is the same: given two lists of terms from a similar domain in two different languages, we would like to align the terms in the two lists to get one bilingual glossary to be used in a variety of settings (computer-assisted translation, machine translation, ontology creation etc.). We followed the approach described above faithfully except in the following cases:

- We are focusing only on the English-Slovenian language pair.
- We use newer datasets. The Eurovoc thesaurus currently contains 7083 terms. Similarly, the DGT translation memory contains additional content not yet present in 2013.
- Because our languages (English, Slovenian) don't have compounds, we are not utilizing the approach to compounding described by Aker et al. (2013) for German and some other languages.
- Since no particular cleaning of training data (e.g., manual removal of specific entries) is described in the paper for Slovene, we do not perform any.

We don't think these differences are significant and the experiments should yield similar results.

3.1. Problems with reimplementation

While the general approach is clearly laid out in the article, there are several spots where further clarification would be welcome:

- There is no information about the Giza++ settings or whether the input corpora have been lemmatized. In order to improve term matching, we experimented with and without lemmatization of the Giza++ input corpora.
- There is no information about the specific character mappings rules other than a general principle of one character in the source being mapped to one or more character in the target. Since the authors cover 20 languages, it is understandable that the they cannot include the actual mapping rules in the article. Therefore, we have created our own mapping rules for English-Slovenian according to the instructions in the original paper:
  - Mapping the English term to the Slovenian writing system (the character before the colon is replaced by the sequence of characters after the colon): *x:ks, y:j, w:v, q:k*
  - Mapping the Slovenian term to the English writing system: *č:ch, š:sh, ž:zh*
- We believe that the formula for the Needleman-Wunsch distance in the paper is wrong: instead of $\frac{LCST}{min[len(source)+len(target)]}$ it should be $\frac{LCST}{min[len(source),len(target)]}$ as in Nassirudin and Purwarianti (2015).

We contacted the original authors of the paper and did receive some answers confirming our assumptions (e.g. regarding mapping terms to the different writing systems and that the test set data was selected individually for each language pair), but several other issues remained unaddressed (in particular, what was the exact train and test data selection strategy for the EN-SL language pair). Further inquiries proved unsuccessful due to time constraints on the part of the original authors. We think one of the reasons the lack of clarity of the original paper is its scope: they deal with more than 20 language pairs and it would be impossible to specific information regarding all of them. However, the fact that they deal with all Eurovoc language pairs is also one of the strengths of the original paper.

3.2. Results

The evaluation on the test set of 1,416,600 English-Slovene term pairs shows that compared to the results reported by Aker et al. (2013) (see line 1 in Table 3), our results are significantly worse. Despite all our efforts to follow the original approach, we were unable to match the results achieved in the original paper when running the algorithm without any changes to the original approach. When trying to follow the original paper's methodology, precision is only 3.59% and recall is 88.00% (see line 2 in Table 3 for details.) In addition to 526 positive examples (out of a total of 600), the classifier returns also 14,194 misclassified examples - incorrect term pairs wrongly classified as correct. We have performed an error analysis and found that almost all incorrectly classified term pairs are cases of partial translation where one unit in a multi-word term has a correct Giza++ dictionary translation in the corresponding term in the other language (Some examples can be seen in Table 2). Such examples accounted for around 82% (11,663) of misclassified term pairs.

4. Adaptation: Experiments for improving the reimplementation

The results in our experiments differ dramatically from the results obtained by Aker et al. (2013). Their approach yields excellent results with perfect precision (100%) and 66% recall for Slovenian. Given that there are 600 positive term pairs in the test set, their results mean that the classifier returns only around 400 term pairs. In contrast, our reimplementation attempts saw the classifier return a lot more of total assigned positive term pairs - 14,720, with 14,194 of them misclassified (false positives).

Feature	Category	Description	Type
isFirstWordTranslated	Dictionary	Checks whether the first word of the source term is a translation of the first word in the target term	Binary
isLastWordTranslated	Dictionary	Checks whether the last word of the source term is a translation of the last word in the target term	Binary
percentageOfTranslatedWords	Dictionary	Ratio of source words that have a translation in the target term	Numeric
percentageOfNotTranslatedWords	Dictionary	Ratio of source words that do not have a translation in the target term	Numeric
longestTranslatedUnitInPercentage	Dictionary	Ratio of the longest contiguous sequence of source words which has a translation in the target term (compared to the source term length)	Numeric
longestNotTranslatedUnitInPercentage	Dictionary	Ratio of the longest contiguous sequence of source words which do not have a translation in the target term (compared to the source term length)	Numeric
Longest Common Subsequence Ratio	Cognate	Measures the longest common non-consecutive sequence of characters between two strings	Numeric
Longest Common Substring Ratio	Cognate	Measures the longest common consecutive string (LCST) of characters that two strings have in common	Numeric
Dice similarity	Cognate	2*LCST / (len(source) + len(target))	Numeric
Needlemann-Wunsch distance	Cognate	LCST / min(len(source), len(target))	Numeric
Normalized Levensthein distance (LD)	Cognate	1 - LD / max(len(source), len(target))	Numeric
isFirstWordCovered	Combination	A binary feature indicating whether the first word in the source term has a translation or transliteration in the target term	Binary
isLastWordCovered	Combination	A binary feature indicating whether the last word in the source term has a translation or transliteration in the target term	Binary
percentageOfCoverage	Combination	Returns the percentage of source term words which have a translation or transliteration in the target term	Numeric
percentageOfNonCoverage	Combination	Returns the percentage of source term words which have neither a translation nor transliteration in the target term	Numeric
difBetweenCoverageAndNonCoverage	Combination	Returns the difference between the last two features	Numeric

Table 1: Features used in the experiments. Note that some features are used more than once because they are direction-dependent.

EN	SL	Giza++
agrarian reform	kmetijski odpadki	agrarian, kmetijski, 0.29737
Brussels region	območje proste trgovine	region, območje, 0.0970153
energy transport	nacionalni prevoz	transport, prevoz, 0.442456
fishery product	tekstilni izdelek	product, izdelek, 0.306948

Table 2: Examples of negative term pairs misclassified as positive. Column 1 contains the English term, column 2 contains the Slovenian term and column 3 contains the Giza++ dictionary entry responsible for positive dictionary-based features.

These results are clearly not useful for our goals, which is why in this section we present several methods aiming at improving the results. To do so, we have taken the following steps:

- Giza++ cleaning
- Lemmatization
- Using only those terms that can be found in the Giza++ training corpora (i.e. DGT)
- Same ratio of positive/negative examples in the training and test set
- Training set selection

4.1. Giza++ cleaning

The output of the Giza++ tool contained a lot of noise and we thought it could perhaps have a detrimental effect on the results. There is no mention of any sophisticated Giza++ dictionary cleaning in the original paper beyond removing all entries where probability is lower then 0.05 and entries where the source word is less than 4 characters and the target word more than 5 characters in length and vice versa. For clean Giza++ dictionaries, we used the resources described in Aker et al. (2014) and available via the META-SHARE repository[4] (Piperidis et al., 2014), specifically, the

[4] http://metashare.tilde.com/repository/browse/probabilistic-

No.	Config	Training set size	Training set pos/neg ratio	Precision	Recall	F-score
1	Reported by (Aker et al., 2013)	12,400	1:1	1	0.6600	0.7900
2	Reimplementation approach	12,966	1:1	0.0359	0.8800	0.0689
3	Giza++ cleaning	12,966	1:1	0.0384	0.7789	0.0731
4	Giza++ cleaning and lemmatization	12,966	1:1	0.0373	0.8150	0.0713
5	Only terms that are in Giza++	8,306	1:1	0.0645	**0.9150**	0.1205
6	Training set 1:200	1,303,083	1:200	0.4299	0.7617	0.5496
7	Training set filtering 1	6,426	1:1	0.5969	0.64167	0.6185
8	Training set filtering 2	35,343	1:10	0.9042	0.5350	**0.6723**
9	Training set filtering 3	645,813	1:200	**0.9342**	0.4966	0.6485

Table 3: Results. No. 1 presents the results reported by the authors, No. 2 our reimplementation of the approach and No.3-9 our modifications of the first reimplementation with the aim of improving the results.

transliteration-based approach which yielded the best results according to the cited paper.
With clean Giza++ dictionaries, precision and F-score improves marginally by less than 0.5% at a cost of a much lower recall (10% lower). For details, see Table 3, line 3.

4.2. Lemmatization

The original paper does not mention lemmatization which is why we assumed that all input data (Giza++ dictionaries, EUROVOC thesaurus) are not lemmatized. They state that to capture words with morphological differences, they don't perform direct string matching but utilize Levenshtein Distance and two words are considered equal if the Levenshtein Distance (Levenshtein, 1966) is equal or higher than 0.95. This led us to believe that no lemmatization was used. Nevertheless, we thought lemmatizing the input data could potentially improve the results which is why we adapted the algorithm to perform lemmatization (using Lemmagen (Juršic et al., 2010)) of the Giza++ dictionary and the EUROVOC terms. We have also removed the Levensthein distance string matching and replaced it with direct string matching (i.e. word A is equal to word B, if word A is exactly the same as B), which drastically improved the execution time of the software.
We considered lemmatization as a factor that could explain the difference in results obtained by us and Aker et al. (2013), but our experiments on lemmatized and unlemmatized clean Giza++ dictionaries show that lemmatization does not have a significant impact on the results. Compared to the configuration with unlemmatized clean Giza++ dictionaries, in the configuration with lemmatized Giza++ dictionaries precision was slightly lower (by 0.1%), recall was a bit higher (by around 4%) and F-score was lower by 0.2%. For details, see Table 3, line 4.

4.3. Using only those terms that can be found in the Giza++ training corpora

We thought that one of the reasons for low results can be that not all EUROVOC terms actually appear in the Giza++ training data (i.e. DGT translation memory). The term translations that do not appear in the Giza++ training data could have dictionary-based features similar to the generated negative examples, which could affect the precision of a classifier that was trained on those terms. We found that only 4,153 out of 7,083 terms of the entire EUROVOC thesaurus do in fact appear in a DGT translation memory. Using only these terms in the classifier training set, did improve the precision to 6.5% and recall to 91.5%. For details, see Table 3, line 5.

4.4. Unbalanced training set

In the original paper, the training set is balanced (i.e. the ratio of positive vs. negative examples is 1) but the test set is not (the ratio is around 1:2000). Since our classifier had low precision and relatively high recall, we figured that an unbalanced training set with much more negative than positive examples could improve the former. To test this, we trained the classifier on an unbalanced training set with a 1:200 ratio of positive vs. negative examples [5] This improved precision of the classifier to 42.99% but reduced recall to 76.16%. Nevertheless, we managed to improve the F-score from 6.9% in the reimplementation approach to 54.9%. For details, see Table 3, line 6.

4.5. Training set filtering

The original paper mentions that their classifier initially achieved low precision on Lithuanian language training set, which they were able to improve by manually removing 467 positive term pairs that had the same characteristics as negative examples from the training set. No manual removal is mentioned for Slovenian.
According to our error analysis, the main problem present partial translations in positive term pairs, where one of the words in the source term has a corresponding translation in the target term. These terms have similar characteristics as a number of generated negative examples, which are consequently classified as false positives. To solve this problem, we focused on the features that would eliminate this partial translations from the training set. After a systematic experimentation, we noticed that we can drastically improve precision if we only keep positive term pairs with the following feature values in the training set:

bilingual-dictionaries-from-dgt-parallel-corpus-for-slovenian-english/fa1e1cb47ef111e5aa3b001dd8b71c66f763b373c00545dfb239b12751e5b339/

[5] 1:200 imbalance ratio was the largest imbalance we tried, since the testing results indicated that no further gains could be achieved by increasing the imbalance even more

- isfirstwordTranslated = True
- islasttwordTranslated = True
- percentageOfCoverage > 0.66
- isfirstwordTranslated-reversed = True
- islasttwordTranslated-reversed = True
- percentageOfCoverage-reversed > 0.66

We managed to improve precision to 59.7% with this approach (see Table 3, line 7.) and when combining it with the previous approach of having an unbalanced training set, we manage to achieve a 90.42% precision and a 53.50% recall, improving the F-score to 67.23% (see Table 3, line 8), when the imbalance ratio was 1:10. With an even more unbalanced training set (1:200), we managed to achieve the best precision of 93.42% at the expanse of a lower recall (49.43%).

5. Manual evaluation

Similar to the original paper, we also performed manual evaluation on a random subset of term pairs classified as positive by the classifier (using the configuration No. 9 that yielded the best results). While the authors of the original approach extract monolingual terms using the term extraction and tagging tool TWSC (Pinnis et al., 2012), we use a terminology extraction workflow described in Vintar (2010) and further expanded in Pollak et al. (2012). Both use a similar approach - terms are first extracted using morphosyntactic patterns and then filtered using statistical measures: TWSC uses pointwise mutual information and TF*IDF, while Vintar (2010) compares the relative frequencies of words composing a term in the domain-specific (i.e. the one we are extracting terminology from) corpus and a general language corpus.

In contrast to the original paper where they extracted terms from domain-specific Wikipedia articles (for the English-German language pair), we are using two translation memories - one containing finance-related content, the other containing IT content. Another difference is that extraction in the original papers was done on comparable corpora, but we extracted terms from parallel corpora - which is why we expected our results to be better. Each source term is paired with each target term (just as in the original paper - if both term lists contained 100 terms, we would have 10,000 term pairs) and extract the features for each term pair. The term pairs were then presented to the classifier that labeled them as correct or incorrect term translations. Afterwards, we took a random subset of 200 term pairs labeled as correct and showed them to an experienced translator[6] fluent in both languages who evaluated them according to the criteria set out in the original paper:

- **1 - Equivalence**: The terms are exact translations/transliterations of each other.
- **2 - Inclusion**: Not an exact translation/transliteration, but an exact translation/transliteration of one term is entirely contained within the term in the other language.
- **3 - Overlap**: Not category 1 or 2, but the terms share at least one translated/transliterated word.
- **4 - Unrelated**: No word in either term is a translation/transliteration of a word in the other.

Domain	1	2	3	4
Reported in Aker et al. (2013)				
IT, Ann. 1	0.81	0.06	0.06	0.07
IT, Ann. 2	0.83	0.07	0.07	0.03
Auto, Ann. 1	0.66	0.12	0.16	0.06
Auto, Ann. 2	0.60	0.15	0.16	0.09
Reimplementation				
Finance	0.72	0.09	0.12	0.07
IT	0.79	0.01	0.09	0.12

Table 4: Manual evaluation results. Ann. stands for "Annotator" since the original paper uses two annotators.

The results of the manual evaluation can be found in Table 4. Manual evaluation showed that 72% of positive term pairs in the Finance domain, and 79% of positive term pairs in the IT domain were correctly classified by the classifier. Compared to the original paper, we believe these results are comparable when taking into account the different monolingual extraction procedures ((Pinnis et al., 2012) vs. (Vintar, 2010)), the different language pairs (English-German vs. English-Slovenian) and the human factor related to different annotators. Note however, that given the fact that we used parallel corpora, we would expect our results to be better.

6. Conclusions and future work

In this paper, we tried to reimplement the approach to bilingual term alignment using machine learning by Aker et al. (2013). They approach term alignment as a bilingual classification task - for each term pair, they create various features based on word dictionaries (i.e. created with Giza++ from the DGT translation memory) and word similarities across languages. They evaluated their classifier on a held-out set of term pairs and additionally by manual evaluation. Their results on the held-out set were excellent, with 100% precision and 66% recall for the English-Slovenian language pair.

Our reimplementation attempt focused just on the English-Slovenian language pair (in contrast with the original article where they had altogether 20 language pairs) and we were unable to replicate the results following the procedures described in the paper. In fact, our results have been dramatically different from the original paper with precision being less than 4% and recall close to 90%. We then tested several different strategies for improving the results ranging from Giza++ dictionary cleaning, lemmatization, different ratios of positive and negative examples in the training and test

[6]The original paper used two annotators, hence two lines for each domain in Table 4

sets, to training set filtering. The last strategy proved to be the most effective - we were able to achieve a precision of almost 91% and a recall of 52% which is closer to the original results reported by the authors of the approach. It is possible that in the original experiments authors performed a similar training set filtering strategy, because the original paper mentions that their classifier initially achieved low precision on Lithuanian language training set, which they were able to improve by manually removing positive term pairs that had the same characteristics as negative examples from the training set. However, no manual removal is mentioned for Slovenian. We have also performed manual evaluation similar to the original paper and reached roughly the same results.

This paper demonstrates some of the obstacles for research reimplementation, such as lack of detail and code unavailability. We believe that in this particular case, the discrepancy in the results could be attributed to the scope of the original paper - with more than 20 languages which is also a demonstration of very impressive approach, it would be impossible to describe procedures for all of them. We weren't able to replicate the results of the original paper, but after developing the optimization approaches describe above over the course of several weeks, we were able to reach a useful outcome at the end. We believe that, when the scope of the paper is broad, providing supplementary material online, and preferably the code, is the only was on assuring complete replicability of results. For this reason, in order to help with any future reimplementation attempts of our paper, we are publishing the code at: `http://source.ijs.si/mmartinc/4real2018`.

In terms of future work, we will continue working on improving the accuracy of the classifier, by incorporating the features derived from the parallel corpora (e.g. co-frequency and other measures, see Baisa et al. (2015)), since our main interest is in aligning terminology from translation memories.

7. Acknowledgements

The authors acknowledge the financial support from the Slovenian Research Agency for research core funding (No. P2-0103). The research was partly supported by the industrial project TermIolar (2015-2017)).

8. Bibliographical References

Aker, A., Paramita, M., and Gaizauskas, R. (2013). Extracting bilingual terminologies from comparable corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 402–411.

Aker, A., Paramita, M. L., Pinnis, M., and Gaizauskas, R. (2014). Bilingual dictionaries for all eu languages. In *LREC 2014 Proceedings*, pages 2839–2845. European Language Resources Association.

Baisa, V., Ulipová, B., and Cukr, M. (2015). Bilingual terminology extraction in sketch engine. In *Ninth Workshop on Recent Advances in Slavonic Natural Language Processing*, pages 61–67.

Baldwin, T. and Tanaka, T. (2004). Translation by machine of complex nominals: Getting it right. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*, pages 24–31. Association for Computational Linguistics.

Cao, Y. and Li, H. (2002). Base noun phrase translation using web data and the em algorithm. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Daille, B. and Morin, E. (2005). French-english terminology extraction from comparable corpora. In *International Conference on Natural Language Processing*, pages 707–718. Springer.

Foo, J. (2012). *Computational terminology: Exploring bilingual and monolingual term extraction*. Ph.D. thesis, Linköping University Electronic Press.

Gaizauskas, R., Aker, A., and Yang Feng, R. (2012). Automatic bilingual phrase extraction from comparable corpora. In *24th International Conference on Computational Linguistics*, page 23.

Joachims, T. (2002). *Learning to classify text using support vector machines: Methods, theory and algorithms*. Kluwer Academic Publishers.

Juršic, M., Mozetic, I., Erjavec, T., and Lavrac, N. (2010). Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16(9):1190–1214.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February.

Macken, L., Lefever, E., and Hoste, V. (2013). Texsis: Bilingual terminology extraction from parallel corpora using chunk-based alignment. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 19(1):1–30.

Mann, G. S. and Yarowsky, D. (2001). Multipath translation lexicon induction via bridge languages. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

Nassirudin, M. and Purwarianti, A. (2015). Indonesian-japanese term extraction from bilingual corpora using machine learning. In *Advanced Computer Science*

and Information Systems (ICACSIS), 2015 International Conference on, pages 111–116. IEEE.

Och, F. J. and Ney, H. (2000). A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 1086–1090. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Pinnis, M., Ljubešic, N., Stefanescu, D., Skadina, I., Tadic, M., and Gornostay, T. (2012). Term extraction, tagging, and mapping tools for under-resourced languages. In *Proceedings of the 10th Conference on Terminology and Knowledge Engineering (TKE 2012), June*, pages 20–21.

Piperidis, S., Papageorgiou, H., Spurk, C., Rehm, G., Choukri, K., Hamon, O., Calzolari, N., Del Gratta, R., Magnini, B., and Girardi, C. (2014). Meta-share: One year after. In *LREC*, pages 1532–1538.

Pollak, S., Vavpetic, A., Kranjc, J., Lavrac, N., and Vintar, S. (2012). Nlp workflow for on-line definition extraction from english and slovene text corpora. In *KONVENS*, pages 53–60.

Steinberger, R., Pouliquen, B., and Hagman, J. (2002). Cross-lingual document similarity calculation using the multilingual thesaurus eurovoc. *Computational Linguistics and Intelligent Text Processing*, pages 101–121.

Vintar, S. (2010). Bilingual term recognition revisited: The bag-of-equivalents term alignment approach and its evaluation. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 16(2):141–158.

A 20% Jump in Duplicate Question Detection Accuracy ? Replicating IBM team's experiment and finding problems in its data preparation

João Silva, João Rodrigues, Vladislav Maraev, Chakaveh Saedi and António Branco
University of Lisbon
NLX-Natural Language and Speech Group, Department of Informatics
Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa
Campo Grande, 1749-016 Lisboa, Portugal
{jsilva, joao.rodrigues, vlad.maraev, chakaveh.saedi, antonio.branco}@di.fc.ul.pt

Abstract

Validation of experimental results through their replication is central to the scientific progress, in particular in cases that may represent important breakthroughs with respect to the state of the art. In the present paper we report on the exercise we undertook to replicate the central result of the experiment reported in the Bogdanova et al. (2015) paper, *Detecting Semantically Equivalent Questions in Online User Forums*, which achieved results far surpassing the state-of-the-art for the task of duplicate question detection. In particular, we report on how our exercise allowed to find a flaw in the preparation of the data used in that paper that casts justified doubt on the validity of the breakthrough results reported there.

Keywords: replication, duplicate question detection, convolutional neural network

1. Introduction

This paper reports on the replication of the research results reported in *Detecting Semantically Equivalent Questions in Online User Forums* (Bogdanova et al., 2015), a paper published in the proceedings of the 19th Conference on Computational Natural Language Learning (CoNLL) and henceforth referred to as DSEQ.

The DSEQ paper caught our attention — and the attention of everyone doing research on Duplicate Question Detection, we guess — by reporting an accuracy of over 92% on the detection of semantically equivalent questions. This is an accuracy score that was more than 15 points above the results achieved in the related literature for that task (Nakov et al., 2016, Task B), representing a notable progress of 20% with respect to the state of the art.

This result placed DSEQ at the forefront of the research on Duplicate Question Detection (DQD). As such, while driven to advance our understanding of DQD and to improve its application, we considered the replication of DSEQ as being essential to our research on this topic.

The DQD task consists of classifying two input interrogatives sentences on whether they are a duplicate of each other, as in the following example:

(A) Can I install Ubuntu and Windows side by side?

(B) How do I dual boot Windows along side Ubuntu?

Two questions are semantically equivalent if they can be adequately answered by the same answer.

DQD belongs to the family of Semantic Text Similarity (STS) tasks, which assess the degree to which two textual segments are semantically similar. While the DQD task deals with the semantic equivalence of interrogative sentences, it is implicitly understood that the broader STS tasks concern only declarative sentences.

STS and DQD are language processing tasks of the utmost importance, both having been addressed in SemEval competitive shared tasks: STS since 2012 (Agirre et al., 2012) and DQD since 2016 (Nakov et al., 2016). Both tasks have been useful to support conversational interfaces and chatbots, in general, and online question & answering (Q&A) community forums, in particular.

One of the challenges faced by Q&A communities (online user forums) is that different users at different times frequently post duplicate questions that have already been answered before, rendering the forum potentially inefficient and demanding time-consuming human moderation to flag such duplicates. Adopting an automated system that can detect duplicate questions provides a computational technique to mitigate or solve this issue.

Section 2 provides an overview of the original DSEQ paper. The replication effort, described in Section 3, turned out to be a challenging task, requiring full attention to details in the implementation of the classifiers to work out unreported assumptions. We conclude that the replication can be successfully achieved but only if the segments in the pairs being classified already contain information as to their status as reciprocal duplicates, thus affecting the validity of the results reported in DSEQ.

Additionally, we report on experiments we undertook after we cleaned the data from these indications of the status of the segments in the pairs, which obtained results in the range of the state of the art results reported in the literature for other approaches and methods.

We address the resulting implications along with other considerations in Section 4.

2. The DSEQ paper

This section provides an overview of DSEQ. For the full details, we direct the reader to the original paper.

The DSEQ paper "aims to detect semantically equivalent questions in online user forums". The authors follow the usual definition used in the field of DQD according to which two questions are considered to be semantically equivalent if they can be adequately answered by the same answer.

DQD systems usually resort to supervised machine learning methods, which require labeled data to train a model. For DQD, these data consist of several pairs of text segments with each pair (i.e. the two questions) being labeled as either containing duplicate or non-duplicate segments.

DSEQ implemented different supervised machine learning systems as classifiers in a DQD task by resorting to pairs of questions annotated as duplicates or not duplicates. The aim of machine learning algorithms is to generalize over the training data, in our case, a semantic generalization that is aimed to classify testing data — unseen pairs of questions — and correctly assesses if they are semantically equivalent, i.e. duplicates.

This is a challenging task given that all questions can be rephrased in multiple ways. However, the recent boost in the amount of available data and computational power supported the application of machine learning techniques, including neural network models. This motivated DSEQ to compare standard machine learning methods and a convolutional neural network on a DQD task.

In the next subsections, we briefly describe the data used for the training of the machine learning models, the machine learning methods resorted to, the experiments performed, and the results reported.

2.1. Data sets used

Stack Exchange[1] is one of the largest Q&A online communities, with over 100 million monthly unique visitors. Like in all Q&A online communities, users can ask questions, get answers from other members of the community, and use a Q&A search engine to find existing questions. Stack Exchange is organized in such a way that each question consists of a *title* (usually a short, one-sentence formulation of the question) and a *body* that provides further details; these are followed by a *thread* of possible answers, ranked by the community.

Stack Exchange allows its users to tag posted questions as duplicates of previously posted questions. These tagged questions are later manually verified by moderators and definitely labeled as duplicates or not. If a question is marked as a duplicate of an already existing one, the moderators may choose to keep it as a duplicate and link it to that pre-existing question. In this way, duplicate questions (i.e. the different ways of asking the same question) end up linked to one and only one canonical formulation for that question.

DSEQ used the data from two Stack Exchange subcommunities: Ask Ubuntu[2], for users and developers of the Ubuntu operating system, and META Stack Exchange[3], for meta-discussion on issues regarding the Stack Exchange network itself.

[1] https://stackexchange.com/

[2] https://askubuntu.com/

[3] https://meta.stackexchange.com/

The Stack Exchange network provides the user-contributed content from all its Q&A sub-communities by means of publicly available periodic data dumps.[4] The data dumps include the questions (title and body), the answer thread, and meta data regarding each question, in particular its status as a duplicate. Figure 1 shows an example of a duplicate entry.

DSEQ used the Ask Ubuntu data dump from May 2014 and the META Stack exchange dump from September 2014. The instances were randomly selected and class-balanced, resulting in a training/testing sets of 24k/6k pairs for Ask Ubuntu and 20k/4k for META. The validation set is 1k pairs for both. Table 1 summarizes this information.

Data set	Training	Testing	Validation
Ask Ubuntu	24k	6k	1k
META	20k	4k	1k

Table 1: Data sets used in DSEQ for the training, testing and hyper-parameterization optimization (validation set) of the machine learning models. Each instance consists of a pair of questions with a corresponding label (duplicate or non-duplicate).

Regarding data preprocessing, the authors specify that NLTK (Bird et al., 2009) was used for tokenization and that all links were replaced with a unique string.

Each question is taken as a whole, that is, as the concatenation of the title and body parts.

2.2. Methods used for DQD

DSEQ compares (i) a rule-based and traditional similarity measure based on word overlap with shingling (n-grams) and a Jaccard coefficient; (ii) a standard machine learning method, namely Support Vector Machine (SVM); and (iii) a neural network architecture with convolutional layers (CNN).

The **Jaccard coefficient** is computed as a rule-based system. First, a set of n-grams (with n ranging from 1–4) is created from the training data. Second, a Jaccard coefficient for the pairs of questions is computed as

$$J(d_1, d_2) = \frac{S(d_1) \bigcap S(d_2)}{S(d_1) \bigcup S(d_2)},$$

where $S(d_1)$ is the set of n-grams extracted from the first segment (d_1) and $S(d_2)$ the set of n-grams extracted from the second segment (d_2). Segments d_1 and d_2 are deemed to be duplicate if the Jaccard coefficient is above a threshold that is empirically determined by measuring the coefficient of all pairs of questions in the training set.

The **SVM** is a machine learning algorithm that finds a hyperplane that optimizes the division of a data set into two classes. In an SVM, the data set instances are transformed into feature vectors, which are data points in a shared space. Then, a hyperplane is iteratively computed aiming at the best separation of the vectors regarding their classes.

[4] https://archive.org/details/stackexchange

```
1  <row
2      Id="1208"
3      PostTypeId="1"
4      CreationDate="2009-06-30T16:15:07.673"
5      Score="5"
6      ViewCount="152"
7      Body="*"
8      OwnerUserId="130090"
9      LastEditorUserId="-1"
10     LastEditDate="2017-03-20T10:30:01.953"
11     LastActivityDate="2009-07-31T07:48:12.030"
12     Title="Improper pagination of user search"
13     AnswerCount="1"
14     CommentCount="2"
15     ClosedDate="2009-07-19T23:42:02.707"
16 />
17
18 <body>
19     <blockquote>
20     <p>
21         <strong>Possible Duplicate:</strong>
22         <br>
23         <a href='https://meta.stackexchange.com/questions/469/
           page-navigation-on-users-page-doesnt-work-if-using-the-input-box'>
           Page navigation on Users page doesn't work if using the input box
           </a>
24     </p>
25     </blockquote>
26     <p>When searching for a user without knowing the exact name, it shows
       paginated results of all matches. However if you try and navigate
       through these results they are automatically defaulted back to the
       reputation based sort. It's impossible to see any more than 35
       matches for any search.</p>
27     <p>Suggestion:</p>
28     <ol>
29         <li>Either remove the paginated results for user searches </li>
30         <li>Make them work :)</li>
31     </ol>
32 </body>
```

Figure 1: A question data and meta data from the META Stack Exchange dump. For the sake of readability, the HTML entities were normalized and the content of the **Body** attribute (line 7) is shown separately under its own tag `<body>` (lines 18–32).

Resorting to the set of existing n-grams (n ranging from 1–4), for each pair of questions, DSEQ uses a vector with the following features:

1. The one-hot encoding of the n-grams in the first question; that is, for each n-gram, a boolean value indicating its occurrence in the first question.
2. The one-hot encoding of the n-grams in the second question.
3. The overall normalized count of each of the n-grams in both questions.

A radial basis function kernel is used to measure the similarity between feature vectors. The DSEQ's authors mention that a grid search was used to optimize the values of the hyper-parameters C and γ, and a frequency threshold was applied to reduce the features dimension.

A **combination of Jaccard coefficient and SVM** machine learning algorithm was also used. To this purpose, the SVM feature vectors were created as previously described and extended to include an extra field: for each pair, the Jaccard coefficient for that pair was considered in the corresponding feature vector.

A **Convolutional Neural Network** (CNN) is one of many neural network architectures that map an input to an output (class) resorting to layers of connected neurons. Typically, in a neural network each neuron receives input values that are used to output a computed value according to an activation function, such as a binary step or a hyperbolic tangent function. The input values of neurons are usually connections from other neurons. Each connection has an intrinsic weight, used to increase or decrease the values sent through them across neurons, strengthening or weakening the signal.

In the CNN used in DSEQ, each layer is connected consecutively (feedforward), with the output of each layer being sent to the next layer. The neural network receives each of the questions in the pair, with both inputs sharing the same neural network layers, in an architecture called Siamese neural network.

- In a word representation layer, each word of the sentence (question) is transformed into a vectorial representation, also known as a word embedding or distributional semantic vector.
- A convolution layer then computes a new vectorial representation by applying a dimension reduction technique to a matrix populated by all the word vectors from the previous layer. The computation can be observed as a compositional compression, encoding the semantic knowledge of the sentence.
- A final layer compares the representation obtained from both questions, using a cosine similarity function. This value is passed on to an activation function that determines if it is, or not, a duplicate pair.

The neural network learns with the training set to generalize the DQD task by iteratively changing the weights of the neural connections while aiming to output the correct class for each training instance.

2.3. Experiments

Four types of experiments are reported in DSEQ: (i) a comparison of the different DQD methods described above; (ii) an assessment of the impact of using domain-specific distributional semantic vectors; (iii) an assessment of the impact of varying the size of the training set; and (iv) an assessment of performing domain adaptation.

The **comparison of the DQD methods** evaluated each method with different parameterizations over the Ask Ubuntu data set. Two experiments were run, the first with a 4k training set and the second with the full 24k training set. The question title and question body were used as inputs in three different ways, namely (i) using the whole title and body;[5] (ii) removing programming language code snippets; and (iii) prefixing programming language code snippets with a special tag. In all cases, the 1k validation set was used to tune the hyper-parameters of the algorithms.

The **assessment of the impact of using domain-specific distributional semantic vectors** was twofold. It (i) evaluated the accuracy of the CNN using already trained distributional semantic vectors with different dimensions (50, 100, 200 and 400); and (ii) evaluated different distributional semantic vector space trained using Wikipedia data as general domain data, and the Ask Ubuntu data as in-domain data.

The **impact of varying the training set size** was assessed by profiling the different systems using different dataset sizes, from only 100 question pairs to the full 24k question pairs.

The **domain adaptation** experiment interchanged the CNN training data. Different corpora were used for training the machine learning algorithm and the distributional semantic vectors. The evaluation was performed with the META test set.

[5]Taking into account the already mentioned NLTK tokenization and links normalization.

2.4. Results

In the first experiment, when comparing the different DQD systems, the combination of SVM with Jaccard performed better than either of its parts individually. The hybrid system obtained a 77.4% accuracy, with the normalized input (removing data related to programming code), $C = 32.0$ and $\gamma \approx 3.05 \times 10^{-5}$. The CNN obtained the best result, 92.4% accuracy, with the normalized input, a 200 vector dimension, $k = 3$, $cl_u = 300$ and $\gamma = 0.005$.
In the second experiment, the study of the impact of domain-specific distributional semantic vectors, by increasing the vectors dimension, the CNN's accuracy improved. Regarding the use of general domain trained distributional semantic vectors from Wikipedia data against the in-domain Ask Ubuntu ones, the in-domain vectors supported a better accuracy: 85.5% accuracy was obtained with the former and 92.4% with the latter.
The third experiment showed that enlarging the training data improved the accuracy of all the systems.
In the fourth experiment, with the META data set, CNN obtained the best score, 92.68% accuracy, when using the META training data in both the training of the CNN and in the training of the distributional semantic vectors.
In Table 2 the best scores obtained in DSEQ with different data sets are reported.

Data set	Accuracy
Ask Ubuntu	92.90%
META	92.68%

Table 2: The best scores reported in DSEQ for the Ask Ubuntu and the META data sets using CNN.

3. Replication of DSEQ

The present Section describes our replication of the experiments reported in DSEQ paper as providing the best results, just indicated in Table 2.
Neither the data sets nor the software with the implementation of the DQD systems used in DSEQ were made publicly available. We attempted to obtain these data sets and more details about the hyper-parameters of the CNN but our emails received no answer.
When **acquiring the data** for the replication exercise, we realized that the Stack Exchange data dumps are frequently updated, with older data dumps being deleted. Thus, at the time of our exercise, we only had access to data dumps from September 2014, given that data dumps from May 2014 had already been removed from the respective distribution page. Table 3 shows the differences between the dumps used in the present work and in DSEQ.
Our **preparation of the data** — for the training of the CNN and the distributional semantic vectors — comprised the following procedures:

- Image removal.
- URL removal.
- Code snippet removal (i.e. <code> blocks).

Data set	Dump date	
	Replication	DSEQ
Ask Ubuntu	Sep. 2014	May 2014
Meta	Sep. 2014	May/Sep. 2014

Table 3: Dump dates for the data sets used in present replication and in DSEQ.

- Text tokenization, using the Stanford Tokenizer (Manning et al., 2014).
- Lowercasing of all tokens.

For the **training of the distributional vectors** we used the DeepLearning4j toolkit[6] with the built-in skip-gram algorithm.
The vectors were trained with a dimension of 200. The values for all the other parameters, which are not described in the DSEQ paper, were taken from the word2vec vanilla parameters.
Table 4 presents the data sets used for the distributional semantic vectors training.

Data set	Vector size	Types	Tokens
Ask Ubuntu	200	68k	38M
META	200	30k	19M

Table 4: Data sets used in replication to train the distributional vectors and respective sizes.

The **data sets acquired were organized** to approximate the organization of DSEQ data sets by using the same sizes for the training, testing and validation subsets. See Table 5 for a detailed rendering.
The **implementation of the CNN** was done using the Keras Python library (Chollet and others, 2015) with Theano (Team, 2016) as the back-end.
For the CNN hyper-parameters, we used the same values as in the DSEQ, when they were reported. The remaining hyper-parameters, namely batch size and number of epochs, were empirically determined by experimentation.
Table 6 shows the values for the main hyper-parameters used in the CNN replication.
The **evaluation** of the CNN over the Ask Ubuntu data set achieved a 94.1% accuracy, and 94.2% accuracy over the META data set.
The replicated models show a performance that is very similar, or even slightly better, to the one reported in DSEQ. Table 7 collects the relevant scores.

3.1. The problematic clue strings

When preparing the data sets for the replication exercise, we realized that removing the URLs from the data dumps as described in DSEQ was not enough to produce unbiased data sets.
We noticed that duplicate questions contain information that provides explicit *clues* as to their status as a duplicate. In particular, at the start of the `body` content, duplicate questions contain the string `Possible Duplicate:` followed by a link to the canonical question of which the question at stake is a duplicate. This is illustrated in Figure 1, in lines 21–23.
Note that these strings, with this explicit indication of the solution of the DQD task, cannot be left in the data since they provide direct clues for the answer the system should optimally deliver — i.e. whether the questions are duplicate or not.
The replication results we reported above in Table 7 were obtained when such clues were kept in the data.
It is not indicated in the DSEQ paper if these clue strings were kept in or removed from the data sets in the experiment reported therein. But further experiments we undertook provide a strong indication that they were not.
We repeated the same experiments by changing only the way the data sets were prepared, in particular by removing such clue strings from them. The scores obtained in this second round of replication — with data sets cleaned from these clue strings — are in line with the state of the art that existed before the DSEQ paper.
Table 8 presents the comparison of our two replication rounds against DSEQ. When removing the clue strings from all the data sets the accuracy drops in all the experiments; when keeping them, all scores are very close to the ones reported in DSEQ.
This very likely indicates that in the experiments reported in DSEQ the clue strings in to duplicate questions were not removed from the data sets used in its experiments.
The data and models used in the replication exercise reported here are available at this GitHub page.[7]
Due to a couple of implementation details that were left unreported in DSEQ — and we had to figure out by ourselves for the replication exercise (cf. Table 6) — and due to slight differences in the data set dump dates (cf. Table 3), our replication settings are not fully identical to the ones of DSEQ. However, given the results obtained in the different rounds of replication and how they are closely aligned with results from the DSEQ (in the first round) and from the literature (in the second round), these differences are not enough to prevent the main conclusions coming out of the present replication exercise.

4. Conclusions

In the present paper, we describe the exercise we undertook of replicating the experiment described in (Bogdanova et al., 2015), which was reported to outperform by 20% the state of the art in the task of Duplicate Question Detection that was contemporary to the publication of that work.
As in the literature on Duplicate Question Detection the progress reported in different papers typically represent a much smaller delta of progress, this result appeared as an outstanding breakthrough in this area, to which, moreover, none of the subsequent advances reported in the literature

[6] http://deeplearning4j.org/word2vec

[7] `https://github.com/nlx-group/Replication-of-IBM-Team-s-Duplicate-Question-Detection-Experiment`

Data set	Total pairs	Duplicates	Training	Testing	Validation
Ask Ubuntu	167,765	17,115	24k	6k	1k
META	67,746	19,456	20k	4k	1k

Table 5: Splits and sizes of the data sets used in replication to train the CNN.

Parameter	Value	Description
d	200	Size of word representation
k	3	Size of k-gram
cl_u	300	Size of convolutional filter
γ	0.005	Learning rate
batch size	1	Examples per gradient update
epochs	20	Number of Training epochs

Table 6: CNN training hyper-parameters. Only the first four parameters were explicitly provided in DSEQ.

	Data set	Accuracy
DSEQ	Ask Ubuntu	92.90%
	META	92.68%
Replication	Ask Ubuntu	94.10%
	META	94.20%

Table 7: Performance results of DSEQ and of the present replication exercise, using the CNN model for both data sets.

had come close.[8] That was the major motivation for our replication exercise.

The replication exercise reported here permitted to find out that the best scores described in (Bogdanova et al., 2015) can be replicated only when the data sets are not properly prepared. In particular, they can be replicated only when clue strings in the data — with the explicit indication that questions are duplicates — are not removed.

Our replication exercise permitted also to find out that when the data sets are cleaned from these clues, as they should, the accuracy of those very same models drops sharply to scores in line with the state of the art scores reported in the literature contemporary to that paper.

This casts justified doubts on the validity of the breakthrough result reported, indicating a jump of 20% with respect to the state of the art, that does not hold.

The current study also highlights the importance of replication as a first class citizen in research on language technology. If this replication exercise reported here had not be undertaken, the community would have remained with an incorrect believe about what would be the state-of-the-art for the task of Duplicate Question Detection.

[8] Among several others, see the results of SemEval2017, Task 3, Subtask B, reported in (Nakov et al., 2016), and the recent advances obtained by our team, reported in (Rodrigues et al., 2018), (Rodrigues et al., 2017), (Saedi et al., 2017) and (Maraev et al., 2017)

	Ask Ubuntu			META	
Clues	**4k**	**full val.**	**test**	**val.**	**test**
Removed	71.8	73.8	73.3	57.3	55.7
Kept	91.8	92.3	94.1	96.1	94.2
DSEQ	92.4	93.4	92.9	92.8	92.7

Table 8: Accuracy (%) of CNN models over Ask Ubuntu and on META data sets, with clue strings kept and with clue strings removed in replication, compared to DSEQ.

5. Acknowledgments

The present research was partly supported by the Infrastructure for the Science and Technology of the Portuguese Language (CLARIN Língua Portuguesa), by the National Infrastructure for Distributed Computing (INCD) of Portugal, and by the ANI/3279/2016 grant.

6. References

Agirre, E., Gonzalez-Agirre, A., Cer, D., and Diab, M. (2012). Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics (*SEM2012)*, pages 385–393.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Bogdanova, D., dos Santos, C. N., Barbosa, L., and Zadrozny, B. (2015). Detecting semantically equivalent questions in online user forums. In *Proceedings of the 19th Conference on Computational Natural Language Learning (CoNLL2015)*, pages 123–131. http://aclweb.org/anthology/K/K15/K15-1013.pdf.

Chollet, F. et al. (2015). Keras. GitHub.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014)*, pages 55–60.

Maraev, V., Saedi, C., Rodrigues, J., Branco, A., and Silva, J. (2017). Character-level convolutional neural network for paraphrase detection and other experiments. In *Proceedings, 6th Artificial Intelligence and Natural Language Conference*.

Nakov, P., Marquez, L., Moschitti, A., Magdy, W., Mubarak, H., Freihat, A. A., Glass, J., and Randeree, B. (2016). Semeval-2016 task 3: Community question answering. In *Proceedings of the 11th International Conference on Semantic Evaluation (SemEval2016)*, pages 27–48.

Rodrigues, J., Saedi, C., Maraev, V., Silva, J., and Branco, A. (2017). Ways of asking and replying in duplicate question detection. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM2017)*, pages 261–270.

Rodrigues, J., Saedi, C., Branco, A., and Silva, J. (2018). Semantic equivalence detection: Are interrogatives harder than declaratives? In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC2018)*. Accepted, to appear.

Saedi, C., Rodrigues, J., Silva, J., Branco, A., and Maraev, V. (2017). Learning profiles in duplicate question detection. In *Proceedings of the IEEE 18th International Conference on Information Reuse and Integration (IEEE-IRI2017)*.

Team, T. D. (2016). Theano: A python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Examining a hate speech corpus for hate speech detection and popularity prediction

Filip Klubička, Raquel Fernández
School of Computing, Dublin Institute of Technology
Institute of Logic, Language and Computation, University of Amsterdam
filip.klubicka@mydit.ie
raquel.fernandez@uva.nl

Abstract

As research on hate speech becomes more and more relevant every day, most of it is still focused on hate speech detection. By attempting to replicate a hate speech detection experiment performed on an existing Twitter corpus annotated for hate speech, we highlight some issues that arise from doing research in the field of hate speech, which is essentially still in its infancy. We take a critical look at the training corpus in order to understand its biases, while also using it to venture beyond hate speech detection and investigate whether it can be used to shed light on other facets of research, such as popularity of hate tweets.

Keywords: hate speech, machine learning, feature analysis, corpus bias, ephemeral data, replicability

1. Introduction

The Internet, likely one of humanity's greatest inventions, facilitates the sharing of ideas and knowledge, as well as online discussion and user interaction. All these are positive features but, as with any tool, whether they are used in a positive or negative manner depends largely on the people that use them. Consequently, and especially when user anonymity is added to the mix, online discussion environments can become abusive, hateful and toxic. To help identify, study, and ultimately curb this problem, such negative environments and the language used within are being studied under the name *hate speech*.

Research on hate speech has become quite prominent in recent years, with dedicated workshops and conferences,[1] and even being featured on LREC2018's list of hot topics. However, hate speech research is still in its infancy. In part, this is due to the following challenges:

1. The term hate speech is difficult to define. Silva et al. (2016) say that "hate speech lies in a complex nexus with freedom of expression, group rights, as well as concepts of dignity, liberty, and equality. For this reason, any objective definition (i.e., that can be easily implemented in a computer program) can be contested." Generally, the current consensus among researchers seems to be that hate speech can be seen as a phenomenon encompassing issues such as: personal attacks, attacks on a specific group or minority, and abusive language targeting specific group characteristics (e.g., ethnicity, religion, gender, sexual orientation).

2. Creating resources for studying hate speech is far from trivial. Hate speech comprises a very small fraction of online content, and on most social platforms it is heavily moderated. For example, Nobata et al. (2016) report that in their corpus of comments on Yahoo! articles collected between April 2014 and April 2015, the percentage of abusive comments is around 3.4% on Finance articles and 10.7% on News. Since the phenomenon is elusive, researchers often use lists of offensive terms to collect datasets with the aim to increase the likelihood of catching instances of hate speech (Davidson et al., 2017; Waseem and Hovy, 2016). This filtering process, however, has the risk of producing corpora with a variety of biases, which may go undetected.

3. Finally, hate speech is present in user-generated content that is not under the control of the researcher. Social media data is typically collected by public APIs that may lead to inconsistent results. For example, González-Bailón et al. (2014) find that the Twitter Search API yields a smaller dataset than the Stream API when using the same filtering parameters. Furthermore, users might delete their profiles or moderate their own questionable content themselves. Thus, datasets on which research experiments are performed are ephemeral, which makes replication of results very difficult.

In this paper, we focus on the latter two points. We consider a particular hate speech corpus – a Twitter corpus collected by Waseem and Hovy (2016), which has been gaining traction as a resource for training hate speech detection models (Waseem and Hovy, 2016; Gambäck and Utpal, 2017; Park and Fung, 2017) – and analyse it critically to better understand its usefulness as a hate speech resource. In particular, we make the following contributions:

- We report the outcome of a reproduction experiment, where we attempt to replicate the results by Waseem and Hovy (2016) on hate speech detection using their Twitter corpus.
- We use the corpus to study a novel aspect related to hate speech: the popularity of tweets containing hate speech.

[1] A few recent examples: https://europa.eu/newsroom/events/conference-online-hate-speech_en
https://sites.google.com/site/abusivelanguageworkshop2017
http://reportinghate.eu/contact2017/
http://likestiltnorden2017.regjeringen.no/language/en/nordic-hate-speech-conference/

To this end, we develop models for the task of predicting whether a hate tweet will be interacted with and perform detailed feature analyses.

- We perform a quantitative and qualitative analysis of the corpus to analyse its possible biases and assess the generality of the results obtained for the hate speech detection and popularity tasks.

2. Replication: hate speech detection results

We aim to replicate the results on hate speech detection by Waseem and Hovy (2016) using the hate speech Twitter corpus created by the authors.[2] The dataset is a useful resource as it is one of few freely available corpora for hate speech research; it is manually annotated and distinguishes between two types of hate speech – sexism and racism – which allows for more nuanced insight and analysis. Additionally, as a Twitter corpus, it provides opportunity for any type of analysis and feature examination typical for Twitter corpora, such as user and tweet metadata, user interaction, etc.

2.1. Corpus in numbers

Here we provide just a brief quantitative overview of the corpus, whereas a more detailed qualitative analysis is presented in Section 4.. The original dataset contains 16907 annotated tweets. However, as is common practice with Twitter corpora, the corpus was only made available as a set of annotated tweet IDs, rather than the tweets themselves. To obtain the actual tweets and corresponding metadata, we used the Tweepy Twitter API wrapper.[3] Given that the corpus was initially collected and annotated in 2016, there have been some changes in the availability of tweets by the time we extracted in in May 2017. Table 1 presents the distribution of annotations in the corpus in its original version and the version that was used for this paper. A tweet in the corpus can have three labels (None, Racism, Sexism). It is possible that a tweet has multiple labels, in the case that it contains both racism and sexism (this only happens in 8 tweets in the original dataset, so it is not a widespread phenomenon in this corpus.)

Tag	Original	Available	Deleted	Percent
None	11,559	11,104	455	3.94%
Hate	5,340	5,068	222	4.16%
Racism	1,970	1,942	22	1.12%
Sexism	3,378	3,126	200	5.92%
Total	16,907	16,172	735	4.35%

Table 1: Distribution of hate speech annotations in the corpus. Presenting original counts, available counts, the number of unobtainable tweets and the percentage they represent in their respective category.

The dataset is quite unbalanced, but this is reflective of the unbalanced distribution of hate speech 'in the wild', and speaks to why it is so difficult to do research on hate speech in the first place: it is an elusive phenomenon. This, combined with the fact that users might delete their profiles or moderate their own questionable content themselves, makes available data scarce, and makes every Twitter corpus smaller over time, and consequently, less valuable and more prone to mistakes when attempting a replicative study.

2.2. Experimental setup

As with any replication study, our aim here is to mimic the original experimental setup as closely as possible, in hopes of obtaining same or comparable results. Unfortunately, this effort is already potentially hindered by the fact that the Twitter corpus has shrunk over time. However, the difference is not too large, and we expect it not to have a significant impact on the results.

A much more prominent obstacle is the lack of certain implementation details in the original paper that make reproduction difficult. At several points in the pipeline, we were left to our own devices and resort to making educated guesses as to what may have been done, due to the lack of comprehensive documentation. More specifically, there are two important aspects of the pipeline that present us with this problem: the algorithm and the features.

The algorithm. Waseem and Hovy (2016) state that they use a logistic regression classifier for their hate speech prediction task. What is not mentioned is which implementation of the algorithm is used, how the model was fit to the data, whether the features were scaled, and whether any other additional parameters had been used.

Due to its popularity and accessibility, we opt for the Scikit-learn (Pedregosa et al., 2011) Python implementation of the logistic regression algorithm.[4] In addition, after fitting the model, we do not do additional scaling of the features when working with just n-grams (as these are already scaled when extracted), but we do scale our other features using the scaling function.[5]

The features. Waseem and Hovy (2016) explore several feature types: they employ n-gram features – specifically, they find that character n-grams of lengths up to 4 perform best – and in addition, they combine them with gender information, geographic location information and tweet length, finding that combining n-gram features with gender features yields slightly better results than just n-gram features do, while mixing in any of the other features results in slightly lower scores.

As a rule of thumb, we would attempt to replicate the best performing setup (character n-grams in combination with gender). However, this proved to be difficult, as user gender information is not provided by Twitter (hence it cannot be scraped from the Twitter API) and has not been made available by the authors along with their dataset. However, they do describe how they went about procuring the gender information for themselves (by performing semi-automatic, heuristics-based annotation), but only managed to annotate

[2] https://github.com/zeerakw/hatespeech

[3] http://tweepy.readthedocs.io/en/v3.5.0/

[4] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[5] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.scale.html

about 52% of the users. This, in combination with the fact that in the original experiment the F1 score improvement when gender is considered is minor (0.04 points) and not statistically significant, led us to focus our efforts on replicating only the experiments involving n-gram features.
However, extracting the n-gram features is also shown to be a nontrivial task, as the original paper does not state how the features are encoded: whether it is using a bag-of-n-grams approach, a frequency count approach, or a TF-IDF measure for each n-gram. We opt for TF-IDF because it is most informative, and just as easy to implement as the more basic approaches.[6]

2.3. Evaluation and results

The original paper states the use of 10-fold cross-validation for model evaluation purposes, without specifying a particular implementation. For the sake of consistency, we again opt for the Scikit-learn implementation.[7]
We compare the results of our setup to the results of the original experiment. In addition, we also compare evaluations of a system trained on various other features (which we will describe in Section 3.) extracted from the tweets and their metadata. The results are presented in Table 2.

Features	**Original**		**n-grams**		**Other**	
	Acc	F1	Acc	F1	Acc	F1
Regression	-	0.74	0.84	0.71	0.79	0.65

Table 2: Average evaluation scores on the hate speech detection task. The original study only provided an F1 score metric for the logistic regression classifier trained on character n-grams (second column). We replicate this experiment (third column), and also train a logistic regression classifier on the same task (fourth column), but on a different set of features (detailed in Section 3.).

Examining the table reveals that our best attempt at replicating the original experiment, with logistic regression trained only on character n-grams, yields an F1-score that is 0.03 points lower than the original. Such a drop is to be expected, considering that our version of the dataset was smaller and that we had to fill in some gaps in the procedure ourselves, likely resulting in slight procedural mismatches. However, the drop is not large, and might indicate a stable, consistent result.
When looking at the performance of classifiers trained on features extracted from tweets and their metadata, they significantly underperform, with a 6 point drop compared to our replicated experiment, and a 9 point drop compared to the original results. This adds a strong confirmation of an observation made in the original study, namely that n-gram features are the most predictive compared to any other types of features.

[6] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[7] http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

3. New experiment: popularity prediction

To date, most research on hate speech within the NLP community has been done in the area of automatic detection using a variety of techniques, from lists of prominent keywords (Warner and Hirschberg, 2012) to regression classifiers as seen in the previous section (Nobata et al., 2016; Waseem and Hovy, 2016), naive Bayes, decision trees, random forests, and linear SVMs (Davidson et al., 2017), as well as deep learning models with convolutional neural networks (Gambäck and Utpal, 2017; Park and Fung, 2017).
Our intent in this section is to explore hate speech beyond just detection, using the Twitter corpus by Waseem and Hovy (2016). Given that Twitter is a platform that enables sharing ideas, and given that extreme ideas have a tendency to intensely spread through social networks (Brady et al., 2017), our question is: how does the fact that a tweet is a hate tweet affect its popularity?

3.1. Related work

To our knowledge there has not been any work relating tweet popularity with hate speech. However, there is a significant body of work dealing with tweet popularity modeling and prediction. Many papers explore features that lead to retweeting. Suh et al. (2010) perform an extensive analysis of features that affect retweetability, singling out two groups of features: content and contextual features. Similarly, Zhang et al. (2012) train a model to predict the number of retweets using two types of features: user features and tweet features. They also compute information gain scores for their features and build a feature-weighted model. They compare the performance of two algorithms: logistic regression and SVM and find that SVM works better, yielding an F-score of 0.71. In addition, some of the related work also relies on temporal features: Zaman et al. (2013) predict the total number of retweets a given amount of time after posting, using a Bayesian model based on features of early retweet times and follower graphs. Similarly, Hong et al. (2011) predict the number of retweets, using binary and multi-class classifiers. They use a more varied set of features, and aside from temporal features, they use content, topical and graph features, as well as user metadata.
We do not have temporal data at our disposal, nor are we at this stage interested in predicting the exact number of retweets at any given point. We are more concerned with investigating how hate speech comes into play regarding tweet popularity, if at all.

3.2. Popularity analysis

As surveyed above, most of the related work on tweet popularity focuses solely on retweets as indicators of popularity. However, while this is probably the clearest indicator, users can interact with tweets in a number of other ways. For this reason, in the present work we also consider other potential measures of popularity; namely, number of tweet replies and number of 'likes' (formerly called 'favorites').
The number of likes and retweets in the corpus is varied, but highly skewed, with most of the tweets being liked/retweeted 0 times. The distributions are displayed in Tables 3 and 4.

	0	1	2	3	4	5+
Likes	9,393	1,255	246	96	55	59
RTs	10,256	755	54	17	9	13
Replies	10,304	790	7	3	0	0

Table 3: Distribution of the number of interactions on **non-hate speech** tweets constrained to interactions between users in the corpus. Total number of tweets: 11104.

	0	1	2	3	4	5+
Likes	4,696	259	49	27	15	22
RTs	4,857	180	15	6	3	7
Replies	5,049	17	2	0	0	0

Table 4: Distribution of the number of interactions on **hate speech** tweets constrained to interactions between users in the corpus. Total number of tweets: 5068.

Given these distributions, we opt for framing the problem as a binary classification task: we wish to determine whether a tweet receives a reaction (retweet, like, response) at least once, or not at all.

But before we go into prediction, we wish to investigate whether there is a significant difference between hate speech and non-hate speech tweets regarding the number of times a tweet was liked/retweeted/replied to. Thus, to determine whether these differences are statistically significant, we employ the chi-squared (χ^2) statistical significance test. When examining likes and replies, the test yields p-values of $<$0.0001, meaning that tweets containing hate speech in the corpus are both liked and replied to significantly less than non-hate speech tweets are. In other words, if a tweet contains hate speech, it is less likely to be liked and replied to. However, when examining the difference in the number of retweets, the p-value comes out as 0.5967. This means that we cannot dismiss the null hypothesis, or rather, that whether a tweet contains hate speech or not, does not impact its retweetability either way.

3.3. Popularity prediction

Features. We use an large set of features inspired by related work (Waseem and Hovy, 2016; Sutton et al., 2015; Suh et al., 2010; Zaman et al., 2013; Hong et al., 2011; Zhang et al., 2012; Cheng et al., 2014; Ma et al., 2013; Zhao et al., 2015). We divide our features into three groups: Tweet features (metadata about the the tweet itself), user features (metadata about the author of a tweet) and content features (features derived from the content of the tweet), with the largest number of features falling into the latter group. The features are listed in Table 5.

Models and results. We train a logistic regression classifier, as well as a linear SVM classifier to compare their performances. We also train separate models for likes and for retweets. One pair of models was trained on the whole corpus, and two additional pairs of classifiers were trained on just the hate speech portion and non-hate speech portion of the corpus respectively.

We tested all models using 10-fold cross validation, holding out 10% of the sample for evaluation to help prevent overfitting. All modeling and evaluation was performed using Scikit-learn (Pedregosa et al., 2011). The evaluation results are presented in Table 6. We also make our feature dataset, and our training and evaluation scripts available to the community for transparency and reproduction purposes.[8]

Interestingly, our classifiers are consistently better at predicting retweets than likes. Given that they are trained on the same features, this indicates that the nature of these two activities is different, in spite of the fact that they intuitively seem very similar.

Furthermore, it seems that the linear regression model seems to perform slightly better overall than the SVM model on both prediction tasks (likes and retweets).

Tweet features	**User features**
tweet_age	account_age
tweet_hour	len_handle
is_quote_status	len_name
is_reply	num_followers
is_reply_to_hate_tweet	num_followees
num_replies	num_times_user_was_listed
	num_posted_tweets
	num_favorited_tweets
Content features	
is_hate_tweet	has_uppercase_token
has_mentions	uppercase_token_ratio
num_mentions	lowercase_token_ratio
has_hashtags	mixedcase_token_ratio
num_hashtags	blacklist_total
has_urls	blacklist_ratio
num_urls	total_negative_tokens
char_count	negative_token_ratio
token_count	total_positive_tokens
has_digits	positive_token_ratio
has_questionmark	total_subjective_tokens
has_exclamationpoint	subjective_token_ratio
has_fullstop	

Table 5: Features used in the popularity prediction task.

Analysis. In order to investigate which features are most informative for the task, we perform feature ablation according to our feature groups. Some notable results show that removing author metadata from the feature set reduces the performance of the model.[9] However, the biggest takeaway for now is the is_reply feature's impact on the model. Our SVM model's average accuracy drops by 0.04 points if the is_reply feature is omitted from the feature set, whereas omitting many of the other features decreases performance scores by 0.02 points at most, if at all.

Inspired by Zhang et al. (2012), we also calculate infor-

[8]The dataset is comprised of anonymized tweet IDs with extracted content features.

Link to GitHub repository: `https://github.com/GreenParachute/hate-speech-popularity`.

[9]As our analysis in Section 4. will reveal, this seems a consequence of a strong bias towards a handful of overproductive authors in the corpus.

	Whole dataset		Non-hate		Hate	
	Acc	F1	Acc	F1	Acc	F1
Regression						
Likes	0.75	0.57	0.73	0.63	0.79	0.29
Retweets	0.82	0.69	0.81	0.68	0.85	0.73
SVM						
Likes	0.74	0.58	0.73	0.63	0.79	0.16
Retweets	0.81	0.66	0.82	0.69	0.84	0.70

Table 6: Average evaluation scores on binary prediction task (predicting if a tweet will be liked/retweeted or not). Presenting results with different subsets of the corpus and comparing performance of logistic regression and SVM models.

mation gain for all features. The top most informative features for each task (predicting whether a tweet will be liked/retweeted) and for each setup (full dataset/non-hate dataset/hate dataset) according to the information gain (IG) measure are presented in Tables 7, 8 and 9.

Liking		Retweeting	
Feature	IG	Feature	IG
num_tweets	0.1005	is_reply	0.0811
num_followees	0.0999	uppercase_ratio	0.0541
num_liked_tweets	0.0992	has_uppercase	0.0517
num_followers	0.0985	char_count	0.0509
user_id	0.0985	num_tweets	0.0334
account_age	0.098	num_liked_tweets	0.0329
num_listed	0.0954	num_followees	0.0323
len_name	0.0732	num_followers	0.0307

Table 7: Most informative features according to information gain (IG) scores for the **whole dataset**. (A higher score indicates bigger importance.)

Liking		Retweeting	
Feature	IG	Feature	IG
num_followees	0.0648	is_reply	0.1155
num_followers	0.0623	uppercase_ratio	0.0876
num_tweets	0.0622	has_uppercase	0.0877
account_age	0.0605	num_liked_tweets	0.0532
user_id	0.0605	num_listed	0.0529
num_listed	0.0595	num_followers	0.0524
len_handle	0.0552	num_tweets	0.0511
len_name	0.0515	num_followees	0.0495

Table 8: Most informative features according to information gain (IG) scores for the **hate speech subset**. (A higher score indicates bigger importance.)

Given the context of this paper and the nature of the corpus, it is interesting to note that the is_hate_tweet feature does not appear anywhere near the top of the IG rankings, indicating that it is not very informative in regards to predicting whether a tweet will be liked or retweeted.
On a broader note, although the feature lists are more or less

Liking		Retweeting	
Feature	IG	Feature	IG
num_followers	0.0974	is_reply	0.0677
num_followees	0.0970	char_count	0.0519
user_id	0.0963	uppercase_ratio	0.042
account_age	0.0963	has_uppercase	0.0396
num_tweets	0.0953	token_count	0.0323
num_liked_tweets	0.0948	num_followees	0.0258
num_listed	0.0941	num_liked_tweets	0.0257
len_name	0.077	num_followers	0.0246

Table 9: Most informative features according to information gain (IG) scores for the **non-hate speech subset**. (A higher score indicates bigger importance.)

similar across the different dataset splits, there is a marked difference between the retweeting and liking lists, in each split. Features that are very informative for retweeting, but not for liking, are whether the tweet contains uppercase tokens, and, most notably, whether the tweet is a reply. This is in line with our findings in the feature ablation study, confirming that there is a strong link between the possibility of retweeting and whether or not the tweet in question is a reply. Our interpretation of this discrepancy is that original, stand-alone ideas (tweets) might be more likely to be picked up and passed on (retweeted), than a turn in a twitter conversation thread would be. In addition, these overall IG measurements also indicate that there is an inherent qualitative difference between the acts of liking and retweeting.

4. Corpus analysis

As the field of hate speech research is yet to mature, with disagreement about what exactly the phenomenon entails (Waseem et al., 2017) and without a unified annotation framework (Fišer et al., 2017), it is warranted to look at the data and examples in more detail, with considerations for potential shortcomings. In Section 2., we pointed out the ephemeral nature of the corpus by Waseem and Hovy (2016), common to all Twitter datasets. In this section, we analyse other characteristics of the corpus related to the challenges of data collection for hate speech analysis we mentioned in the Introduction (point 2), which can result in undesirable biases.

Tweet collection. Given the small fraction of online content comprised of hate speech, collecting a significant amount of examples is an extremely difficult task. At present, it is not feasible to collect a large sample of tweets and then manually label them as hate or non hate, as the fraction of instances labeled with the positive class will be negligible. The only way to model the phenomenon is to target tweets already likely to contain hate speech.
Driven by this rationale, the authors of the corpus have obtained their dataset by performing an initial manual search of common slurs and terms used pertaining to religious, sexual, gender, and ethnic minorities. The full list of terms they queried for is not very long: *MKR*, *asian drive*, *femi-nazi*, *immigrant*, *nigger*, *sjw*, *WomenAgainstFeminism*, *blameonenotall*, *islam terrorism*, *notallmen*, *victimcard*, *victim card*, *arab terror*, *gamergate*, *jsil*, *racecard*,

race card. In the results obtained from these queries, they identified frequently occurring terms in tweets that contain hate speech and references to specific entities (such as *MKR*, addressed further below). In addition to this, they identified a small number of prolific users from these searches.

This manner of tweet collection allowed the authors to obtain quite a considerable amount of data. However, this approach to data collection inevitably introduces many biases into the dataset, as will be demonstrated further in this section.

Qualitative observations on tweet content. According to the annotation guidelines devised by Waseem and Hovy (2016) for the purpose of annotating this corpus, a tweet is tagged as offensive if it: (1) uses a sexist or racial slur, (2) attacks a minority, (3) seeks to silence a minority, (4) criticizes a minority (without a well founded argument), (5) promotes, but does not directly use, hate speech or violent crime, (6) criticizes a minority and uses a straw man argument, (7) blatantly misrepresents truth or seeks to distort views on a minority with unfounded claims, (8) shows support of problematic hashtags (e.g. #BanIslam, #whoriental, #whitegenocide), (9) negatively stereotypes a minority, (10) defends xenophobia or sexism, (11) the tweet is ambiguous (at best); and contains a screen name that is offensive as per the previous criteria; and is on a topic that satisfies any of the above criteria.

Though at first glance specific and detailed, these criteria are quite broad and open to interpretation. This was likely done to cover as many hate speech examples as possible – a thankless task, as hate speech data is scarce to begin with. However, due to this same breadth, the corpus contains some potential false positives. The most jarring example of this being that, if a user quotes a tweet containing hate speech (by prepending the quoted text with "RT"), the quoter's tweet is tagged as hate speech. Certainly, the user could have quoted the original tweet in support of its message, and even if not, one could argue that they do perpetuate the original hateful message by quoting it. On the other hand, it is just as likely that the user is quoting the tweet not to make an endorsement, but a neutral response. It is even more likely that the user's response is an instance of *counterspeech* — interaction used to challenge hate speech (Wright et al., 2017). Manual inspection shows that there are instances of both such phenomena in the corpus, yet all those tweets are tagged as hate speech. In fact, ∼30% of hate speech tweets in the corpus contain the token 'RT', indicating they are actually retweets. This could pose a problem further down the line when extrapolating information about hate speech users. Addressing this issue would at the very least require going through tweets with quotes and relabeling them, if not altogether rethinking the annotation guidelines, or rather, being more mindful of the semantics at play during annotation.

Topic domain. In spite of the broad guidelines, however, it seems that the actual hate speech examples end up falling on quite a narrow spectrum. Even though the tweets were semi-automatically picked based on a wide variety of keywords likely to identify hate speech, the tag 'racism' is in fact used as an umbrella term to label not only hate based on race/ethnicity, but also religion, specifically Islam. Indeed, the majority of the tweets tagged as racist are, in fact, islamophobic, and primarily written by a user with an anti-Islam handle (as per guideline 11). Though it is stated in the original paper which seed words were used to collect the data (which included both racist and islamophobic terms), it is undeniable that the most frequent words in the racist portion of the corpus refer to islamophobia (which is also explicitly stated by the authors themselves). This is not wrong, of course, but it begs the question of why the authors did not choose a more specific descriptor for the category, especially given that the term 'racism' typically sparks different connotations, ones that, in this case, do not accurately reflect the content of the actual data.

When it comes to sexist tweets, they are somewhat more varied than those annotated as racist. However, they do contain a similar type of bias: ∼13.6% of the tweets tagged as sexist contain the hashtag and/or handle MKR/MyKitchenRules. *My Kitchen Rules* is an Australian competitive cooking game show which is viewed less for the cooking and skill side of the show than for the gossip and conflict which certain contestants are encouraged to cause.[10] It seems to be a popular discussion topic among fans of the show on Twitter, and apparently prompts users to make sexist remarks regarding the female contestants. There is nothing inherently problematic about this being included in a corpus of hate speech, but it cannot be disregarded that more than a tenth of the data on sexism is constrained to an extremely specific topic domain, which might not make for the most representative example of sexism on Twitter.

Distribution of users vs. tweet content Another interesting dimension of the corpus that we observe is the distribution of users in relation to the hate speech annotations – an aspect that could be important for our analysis of popularity presented in Section 3.

There are 1858 unique user IDs in the corpus. Thus many of the 16907 tweets were written by the same people. As a simplistic approximation, we can (very tentatively) label every user that is the author of at least one tweet containing hate speech as a hate user; and users that, in the given dataset, have not produced any tweets containing hate speech we label as non-hate users. Of course, this does have certain drawbacks, as we cannot know that a user does not produce hate speech outside the sample we are working with, but it does provide at least an approximation of a user's production of hate tweets in the sample. Using this heuristic, the distribution of users in the corpus in regards to whether they produce hate speech or not is presented in Table 10.

A really striking discrepancy immediately jumps out when looking at Table 10: there is a total of 5 users responsible for the 1942 racist tweets, as opposed to the 523 users responsible for the 3126 sexist tweets. Assuming normal distribution (which is certainly the wrong assumption), on average there are 388 racist tweets per racist user, while there

[10] https://en.wikipedia.org/wiki/My_Kitchen_Rules#Criticism

User type	Count
Non-hate	1,334
Hate	528
Racist	2
Sexist	520
Racist and sexist	3
Total	1,859

Table 10: Distribution of users according to the type of tweets they produce.

is an average of 6 sexist tweets per sexist user. The actual distribution, however, is extremely skewed – the bulk of all the hate speech data is distributed between three users: one user who produced 1927 tweets tagged as racist, and two users who respectively produced 1320 and 964 tweets tagged as sexist. This is illustrated in Figure 1.

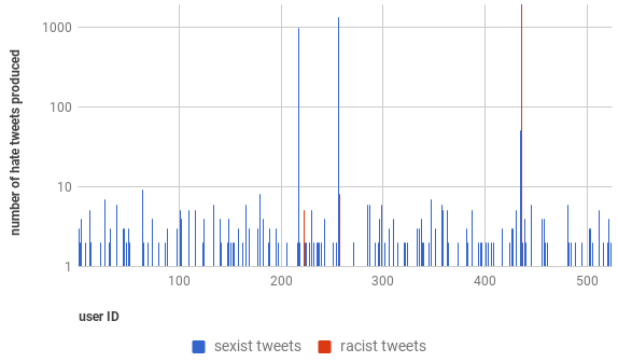

Figure 1: Graph illustrating the distribution of tweets containing hate speech among users producing them. We represent the number of tweets in logarithmic scale.

Such a distribution renders any attempt at generalization or modeling of racist tweets moot, as the sample cannot be called representative of racism as such, but only of the Twitter production of these 5 users.[11] Similarly, the fact that most of the tweets tagged as sexist belong to the same two users considerably skews this subset of the data.

Corollary. All of these points deserve due consideration. The imbalances with respect to distribution of users were certainly considered while we worked with the data. In an attempt to reduce them, we did not distinguish between racist and sexist tweets in our analysis in both Sections 2. and 3. (even though we were tempted to do so), but rather treated them all as simply hate speech tweets. Additionally, it is possible that the insights and biases presented in this section might even call into question the relevance of the findings from Section 3., as the popularity modeled there is likely reflecting the popularity of the particular Twitter users in the corpus rather than of hate speech tweets as such.

[11] However, the data might still be useful when looked at in bulk with sexism, as it might reinforce the similarities they both share stemming from the fact that they are types of hate speech.

5. Conclusion

This paper has provided an overview of several research directions involving hate speech:

1. A critical look at a publicly available hate speech dataset.
2. An attempt at replicating and confirming already established hate speech detection findings.
3. Pushing the research space in a new direction: popularity prediction.

Overall, we analyzed a currently popular hate speech dataset, pointed out considerations that have to be made while working such data, and observed that it is biased on several levels. This does not render it useless, but it is important to keep these biases in mind while using this resource and while drawing any sort of conclusions from the data.

As far as replicability goes, the resource does allow one to model hate speech (as biased as it may be), but not without a certain degree of difficulty. We achieve system evaluation scores of 0.71 in terms of F1 score, which is slightly lower than the original results of 0.74 F1 score on the same setup. The differences and gaps in implementation showcase a common trend in scientific publishing - the general problem of reproducing results due to publications not providing sufficient information to make the experiments they describe replicable without involving guessing games. And only when attempting to reproduce a study can one truly realize how much detail is so easily omitted or overlooked, simply due to lack of awareness.

When it comes to popularity prediction, we determine that hate speech negatively impacts the likelihood of likes and replies, but does not affect likelihood of retweets. However, training only on the hate speech portion of the data does seem to boost our model's performance in retweet prediction. These findings, as well as the evaluation scores and feature analyses, are only the first stepping stone in a long line of future work that can be done to better understand the impact of hate speech on social media and how it spreads. Possibilities include employing social graph mining and network analysis, perhaps using user centrality measures as features in both hate speech and popularity prediction tasks. In addition, reframing the task as not just a binary prediction task, but rather fitting a regression model to predict the exact number of likes, retweets and replies, would certainly be preferable and more informative, and could lead to a better understanding of how hate speech behaves on Twitter.

What is clear is that hate speech is a very nuanced phenomenon and we are far from knowing everything there is to know about it. Resources are scarce and far from perfect, and much more work and careful consideration are needed, as well as much cleaning, fine-tuning, discussion and agreement on what hate speech even is, if we are to build better resources and successfully model and predict hate speech, or any of its aspects.

Acknowledgements

This research was co-funded by the Erasmus+ programme of the European Union and conducted while the first author was visiting the ILLC in Amsterdam. In addition, the

research was supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

6. Bibliographical References

Brady, W. J., Wills, J. A., Jost, J. T., Tucker, J. A., and Van Bavel, J. J. (2017). Emotion shapes the diffusion of moralized content in social networks. *Proceedings of the National Academy of Sciences*, 114(28):7313–7318.

Cheng, J., Adamic, L. A., Dow, P. A., Kleinberg, J. M., and Leskovec, J. (2014). Can cascades be predicted? *CoRR*, abs/1403.4608.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *International AAAI Conference on Web and Social Media*.

Fišer, D., Ljubešić, N., and Erjavec, T. (2017). Legal framework, dataset and annotation schema for socially unacceptable online discourse practices in slovene. In *Proceedings of ALW1: 1st Workshop on Abusive Language Online*.

Gambäck, B. and Utpal, K. S. (2017). Using convolutional neural networks to classify hate-speech. In *Proceedings of ALW1: 1st Workshop on Abusive Language Online*.

González-Bailón, S., Wang, N., Rivero, A., Borge-Holthoefer, J., and Moreno, Y. (2014). Assessing the bias in samples of large online networks. *Social Networks*, 38:16–27.

Hong, L., Dan, O., and Davison, B. D. (2011). Predicting popular messages in twitter. In *Proceedings of the 20th international conference companion on World wide web*, pages 57–58. ACM.

Ma, Z., Sun, A., and Cong, G. (2013). On predicting the popularity of newly emerging hashtags in twitter. *Journal of the American Society for Information Science and Technology*, 64, 07.

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., and Chang, Y. (2016). Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 145–153, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Park, J. H. and Fung, P. (2017). One-step and two-step classification for abusive language detection on twitter. In *Proceedings of ALW1: 1st Workshop on Abusive Language Online*.

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

Silva, L. A., Mondal, M., Correa, D., Benevenuto, F., and Weber, I. (2016). Analyzing the targets of hate in online social media. *CoRR*, abs/1603.07709.

Suh, B., Hong, L., Pirolli, P., and Chi, E. H. (2010). Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *Social computing (socialcom), 2010 ieee second international conference on*, pages 177–184. IEEE.

Sutton, J., Gibson, C. B., Spiro, E. S., League, C., Fitzhugh, S. M., and Butts, C. T. (2015). What it takes to get passed on: message content, style, and structure as predictors of retransmission in the boston marathon bombing response. *PLoS one*, 10(8):e0134452.

Warner, W. and Hirschberg, J. (2012). Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, LSM '12, pages 19–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Waseem, Z. and Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California, June. Association for Computational Linguistics.

Waseem, Z., Davidson, T., Warmsley, D., and Weber, I. (2017). Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of ALW1: 1st Workshop on Abusive Language Online*.

Wright, L., Ruths, D., Dillon, K. P., Saleem, H. M., and Benesch, S. (2017). Vectors for counterspeech on twitter. In *Proceedings of ALW1: 1st Workshop on Abusive Language Online*.

Zaman, T., Fox, E. B., and Bradlow, E. T. (2013). A bayesian approach for predicting the popularity of tweets. *CoRR*, abs/1304.6777.

Zhang, Y., Xu, Z., and Yang, Q. (2012). Predicting popularity of messages in twitter using a feature-weighted model. *International Journal of Advanced Intelligence*.

Zhao, Q., Erdogdu, M. A., He, H. Y., Rajaraman, A., and Leskovec, J. (2015). SEISMIC: A self-exciting point process model for predicting tweet popularity. *CoRR*, abs/1506.02594.

Annotation Contributions: sharing derived research data

Steve Cassidy*, Dominique Estival†,
*Macquarie University, Sydney, Australia,
steve.cassidy@mq.edu.au
†Western Sydney University, Sydney, Australia,
d.estival@westernsydney.edu.au

Abstract

Many research projects make use of language resources and in the process of the research, generate derived resources such as annotations or derived features. These resources are not often shared and there is no mechanism for associating them with the original resource in a way that would help future users discover them. This paper describes a new feature of the Alveo Virtual Laboratory that allows researchers to upload these derived resources and have them associated and integrated with the original resource. This facility is designed to encourage the sharing of these derived resources by making the process of sharing easy and by providing an incentive in the form of a citeable URL that can be properly attributed to the authors in subsequent publications.

Keywords:

1. Introduction

Projects making use of shared language resources often create new artifacts as a by-product of the main research program. These may be annotations of a particular linguistic structure or derived signals such as pitch tracks, translations or summaries of documents. These new resources are often not shared or, if they are, are made available as a separate download not associated with the original language resource. This paper describes a new feature of the Alveo Virtual Laboratory that allows researchers to upload these artifacts and have them associated with the original data they were derived from. These *contributions* are then made available both as a separate resource and linked to the original resource.

The Alveo Virtual Laboratory[1] (Cassidy et al., 2014) combines a data repository with a web-based API and a workflow platform and aims to provide access to data in a way that may enhance reproducibility of research results (Cassidy and Estival, 2017). Data in Alveo is made available as a *collection* of *items* with associated meta-data and one or more *documents* (audio, video, text, annotations, etc). Users can query the data store for items relevant to a particular study, creating an *item list*; this list can then be fed into a data processing pipeline either by downloading data as a zip file or writing scripts against the API that access documents directly. Each item list has a unique URL and can be shared such that other researchers can access the same items (subject to access restrictions) and hence use these in reproducing or extending the original work.

Meta-data in Alveo can be associated with collections, items and documents. The system does not mandate a fixed schema but supports a range of existing vocabularies (OLAC, DCTERMS, etc.). Until recently, all of the data ingested into Alveo has been legacy collections and so the decision was taken to accept any meta-data that was available while providing a mapping to common properties where possible. As a consequence, the Alveo meta-data store is flexible and able to incorporate any meta-data that might be available for a collection (Estival, 2016).

1.1. Sharing Derived Resources

Annotations and other derived resources are often produced as a side-effect of research by language researchers but we are not aware of any repository which accepts resources like these for distribution that associates them in a useful way with the original resource.

As an example of the current state of play in distribution and sharing of annotation we can look at the popular Switchboard corpus published by LDC (Godfrey and Holliman, 1993). The LDC catalog page describing this resource includes a link to the Switchboard Dialog Act Corpus[2], a separate collection of annotations that is distributed as a download from the documentation directory of the corpus on the LDC site. In addition, LDC also publishes the NXT Switchboard Annotations (Calhoun et al., 2009) which combines a number of layers of annotation using the NXT format developed in a collaboration among researchers from Edinburgh University, Stanford University and the University of Washington. On Github we find the *Switchboard Dialog Act Corpus with Penn Treebank links*[3] which builds on the earlier dialog act annotations.

The good part of this story is that we were able to find all of these resources within a few minutes with a few web searches; at least these resources are available on the web and the links are maintained to some degree. However, we can be reasonably sure that there are additional annotations over this data (or corrections to some of these annotations) that are not turning up in our searches - for instance, data that is shared on institutional servers that have a lower page-rank and data that was never shared because the project ran out of funds or had no incentive to publish their data.

There is an opportunity, then, to improve the way that derived resources are published and shared. Language resource providers who make the original datasets available

[1] http://alveo.edu.au/

[2] http://compprag.christopherpotts.net/swda.html

[3] https://github.com/cgpotts/swda

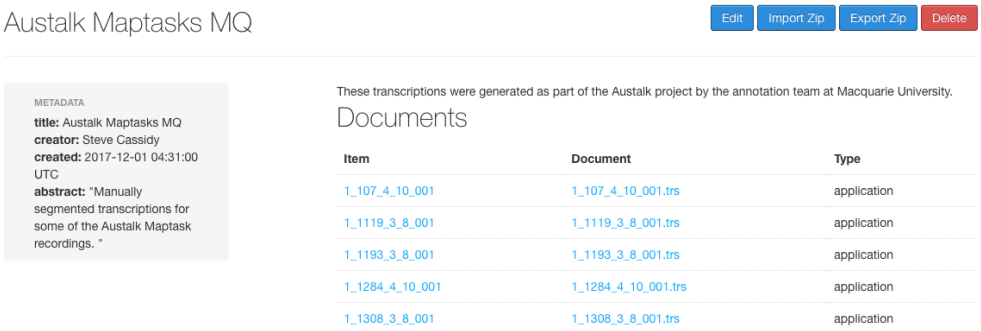

Figure 1: The page describing a contribution.

could develop a way for derived data to be associated with the original resource in such a way that they can be discovered easily by researchers who find the original data.

While most language data archives make collections available as downloadable archives (zip files or similar), Alveo is a more fine-grained store that exposes individual items and documents for discovery and download. Part of the motivation for this is that researchers often only need to use part of a collection in a study; Alveo supports identifying this subset as an *item list* and downloading just that subset of data. This means that we have the opportunity to associate derived resources at this fine-grained level as well. The goal of the work described in this paper is to provide a means for derived resources to be shared on the Alveo platform in such a way that they can be discovered naturally by researchers browsing the original data.

2. Annotation Contributions

The Alveo system has recently been extended to allow users to upload files derived from existing resources and have them associated with the original data they were derived from. This can be illustrated with the following scenario.

A researcher is studying Australian English vowels and identifies 10 speakers from the Austalk corpus, finding the items corresponding to their reading of the 18 hVd words (hid, had, hod, etc). They create an *item list* for these items and download the audio files associated with each. Using a forced aligner they derive TextGrid annotations for each file and hand-correct these to ensure that the vowel boundaries are correctly placed. They then derive formant tracks for each recording and carefully check that the formants are correct, some values are hand-corrected if they have been mis-tracked. This data is then used to derive vowel plots for each of the speakers and the results are written up with reference to speaker meta-data that was downloaded with the original data.

Prior to publication of the study, the researchers want to make the corrected TextGrid and formant files available so that they can be referenced in the paper and made accessible for future researchers. In the Alveo system they create a new *Annotation Contribution* and enter some basic meta-data and a description of the methodology used to create the derived files. The files are then uploaded to Alveo as a zip file. The system unpacks the archive and based on the filenames, locates the original items that they were derived from (e.g. the original WAV file `2_205_1_3_001.wav` was associated with `2_205_1_3_001.TextGrid` and `2_205_1_3_001.fms`).

The new contribution is accessible via a URL which shows a page containing the meta-data and description of the contribution and a list of the items and associated files. From this page, all of the files can be downloaded as a zip file, but the page also shows direct links to the items referenced by the contribution (Figure 2).

The contribution URL is a public page that can be referenced by any web user and hence is suitable for publication in work that references the resources. Users who are not registered with Alveo can see the meta-data and description of the contribution but will not be able to view the associated data or download files without registering and agreeing to the license of the original collection. At this time we do not have any provision for adding an additional license to the contents of the collection. This is something that might be considered in the future.

We are currently able to issue a DOI for *collections* in the Alveo system to facilitate citation of language resources. Since an annotation contribution is an identifiable resource within the system with a unique URL it would also be possible to issue a DOI for a contribution. This is something we will consider as users adopt this new feature.

2.1. Reproducibility

Together with the other facilities of the Alveo platform, this new feature supports a research workflow that can provide for enhanced reproducibility of research outcomes. As described above, the Alveo platform allows the researcher to create an *item list* with the items that are used in an analysis. This list can be shared publicly and cited in the published research. This new feature then allows the derived resources to be shared in the same way, meaning that all of the data that feeds into the analysis in the research can be cited and is available to future researchers. Further, Alveo provides the Galaxy workflow engine (Goecks et al., 2010) as a platform for constructing and running data processing workflows over speech and language data. Galaxy workflows can also be published and cited in such a way that future researchers can reconstruct the exact sequence of the same versions of tools that were used to generate the published results.

However, straight reproduction of results is not the only de-

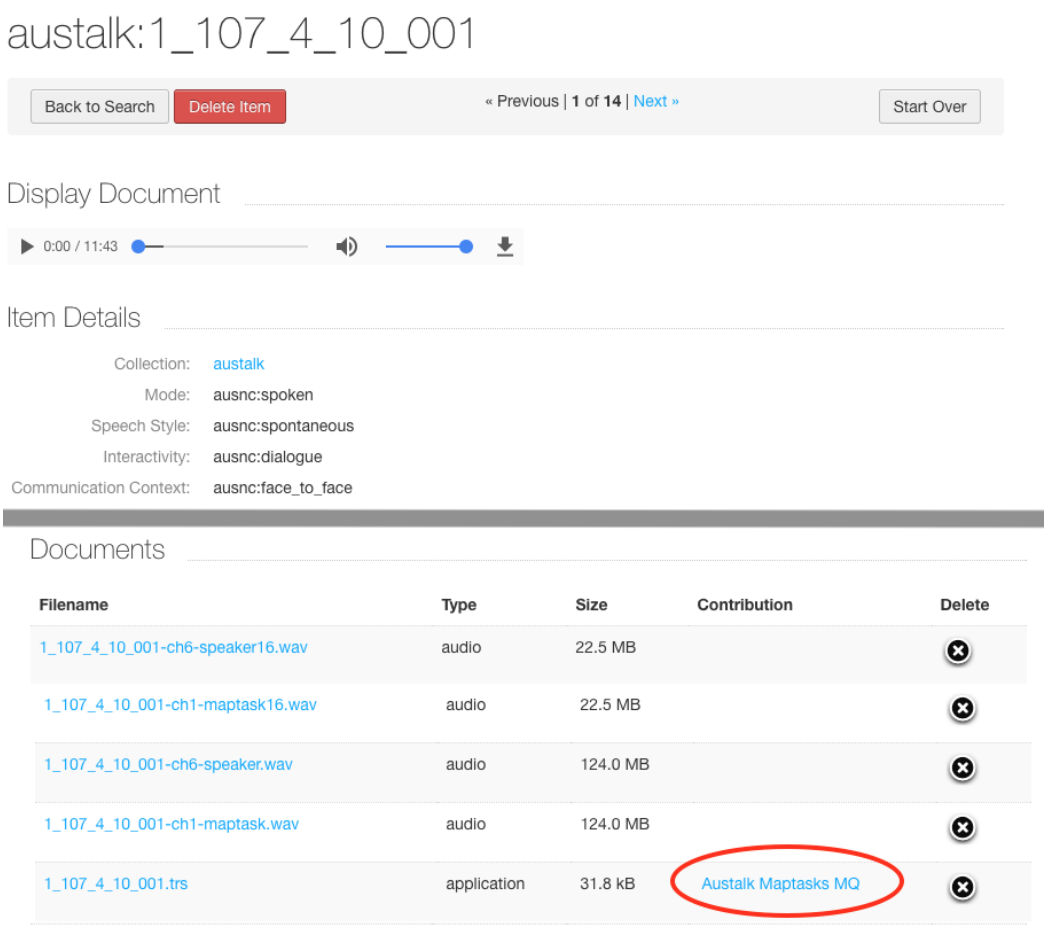

Figure 2: An item page showing the contributed file linking back to the contribution page.

sirable outcome. The publication of derived resources from a research project will also allow future researchers to use this work as a starting point for further investigations. This might be to build on the earlier results or to compare them with results on other data sets. Without easy access to these derived resources, our ability to compare future work with past performance is limited by the detail that is provided in published research procedures. Access to derived resources removes a significant source of variability and can save a lot of effort in repeating earlier analysis.

3. Contributions API

An important part of the Alveo system is the web-based API that provides access to both meta-data and data stored in our collections (Cassidy et al., 2014). Using the API, one can create interfaces to tools for searching and analysing data stored on Alveo. The API has been extended to cover operations on contributions so that external scripts can be written to manipulate them (Table 1).

Using this API, users can write tools as part of an automated annotation workflow to upload the resulting annotation files as part of a new or existing contribution.

`/contrib/`	GET	Get the current list of contributions
`/contrib/`	POST	Create a new contribution from a JSON meta-data description
`/contrib/<id>`	GET	Get the JSON description of a contribution including metadata and a list of document URLs
`/contrib/<id>`	PUT	Update the meta-data for a contribution
`/contrib/<id>`	POST	Add documents to a contribution from a zip file
`/contrib/<id>`	DELETE	Remove a contribution and all associated documents

Table 1: A summary of the contributions API.

4. Provenance Meta-data

The current contribution creation form only provides a very basic set of meta-data fields for the user to complete. The API is able to accept any meta-data fields in the form of JSON-LD formatted properties and values; as mentioned earlier, the Alveo system is able to store arbitrary meta-data structures associated with documents, items, collec-

tions and contributions. One of the goals for future development of the contributions feature is to be able to document the workflow used to generate annotations using the PROV-O provenance ontology.

Belhajjame et al. (2015) describe an extension of the PROV-O ontology for describing scientific workflows. The `wfdesc` ontology allows them to describe a workflow, for example, the processing tools involved and the parameters that they take. The `wfprov` ontology describes the provenance of research artifacts - the execution of workflow steps, inputs and outputs and the parameter settings used in that particular run. These might be used in combination to describe a workflow and the execution settings that generated a set of outputs.

```
{
  "@context": {
     "wfprov":
   "http://purl.org/wf4ever/wfprov#",
     ...
  },
  "@id": "<uri of contribution>",
  "@type": "wfprov:Artifact",
  "wfprov:wasOutputFrom": {
    "@id": "forrestRun13",
    "@type": "wfprov:ProcessRun",
    "wfprov:describedByProcess":
        "toolshed:g_forest/0.01",
    "wfprov:usedInput":
        "<uri of input item list>",
    "wfprov:wasEnactedBy":
        "https://galaxy.alveo.edu.au/",
    "windowShift": 5,
    "windowSize": 20,
    "nominalF1": 500,
    "speakerGender": "Male"
  }
}
```

Figure 3: An example metadata description using the WF-Prov vocabulary derived from PROV-O.

As an example, Figure 3 shows a JSON-LD description of a single processing step that produced a collection of formant tracks using the Emu formant tracker `forest`[4]. The metadata describes the application of the formant tracker via a Galaxy (Goecks et al., 2010) tool (giving the URL of the particular version of the tool repository). In the example, the input is referenced by the URL of an Alveo item list that could be the input to this process; the output is the URL of the contribution itself. Finally, the metadata includes some of the parameter settings used in the execution of this tool.

This kind of metadata could be automatically generated from the provenance data kept by a workflow engine such as Galaxy. Storing this metadata allows a very detailed record to be made of the process used to generate the derived resources. In many cases, the process used to generate derived resources will involve manual steps such as annotation and running interactive tools. An automated capture of the provenance of the output would be difficult in this case, but a user interface for manually entering a structured description following the same format could be built to facilitate recording of this metadata.

5. Adoption by Other Repositories

We have implemented Annotation Contributions in the Alveo system to support the work of researchers working on the data that we hold. The particular design of this feature in our system depends a lot on the other aspects of the system: the individual access to items and documents within collections. However, the idea of encouraging researchers to contribute derived resources back to be associated with the original resource is one that could be adopted by other research data repositories. Even if the data is only available as an archive download, it should be possible to associate new derived datasets with existing collections via their metadata and have them exposed to researchers as they browse the holdings in the repository.

One may go further to develop a standard for linking resources between repositories if the derived resources are, for example, stored in a separate research data store as part of a larger project. As a point of reference, the W3C has recently standardised *Webmentions*[5], a way for websites to notify other sites when their work is mentioned in newly published material. Such a mechanism is built upon the HTTP standard and could be adopted as a means for repositories to notify each other of the availability of derived resources.

6. Summary

This paper has described a new feature of the Alveo Virtual Laboratory that allows researchers to share derived resources generated as part of a research project and have them associated with the original data. This is aimed at improving the sharing of this kind of data that has not been the focus of any other data repository in the past.

7. Acknowledgements

The development of the Alveo Virtual Laboratory was funded by the Australian National eResearch Collaboration Tools and Resources (Nectar) project.

8. Bibliographical References

Belhajjame, K., Zhao, J., Garijo, D., Gamble, M., Hettne, K., Palma, R., Mina, E., Corcho, O., Gómez-Pérez, J. M., Bechhofer, S., Klyne, G., and Goble, C. (2015). Using a suite of ontologies for preserving workflow-centric research objects. *Web Semantics: Science, Services and Agents on the World Wide Web*, 32:16–42, May.

Calhoun, S., Carletta, J., Jurafsky, D., Nissim, M., Ostendorf, M., and Zaenen, A. (2009). NXT Switchboard Annotations LDC2009T26. Web Download, Philadelphia: Linguistic Data Consortium `https://catalog.ldc.upenn.edu/LDC2009T26`.

Cassidy, S. and Estival, D. (2017). Supporting accessibility and reproducibility in language research in the alveo virtual laboratory. *Comput. Speech Lang.*, 45:375–391, September.

[4] `https://github.com/IPS-LMU/wrassp`

[5] `https://www.w3.org/TR/webmention/`

Cassidy, S., Estival, D., Jones, T., Burnham, D., and Burghold, J. (2014). The alveo virtual laboratory: A web based repository API. In Nicoletta Calzolari (conference Chair), et al., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Estival, D. (2016). Alveo: making data accessible through a unified interface – a pipe-dream? In Richard Eckart de Castilho, et al., editors, *Proceedings of the Workshop on Cross-Platform Text Mining and Natural Language Processing Interoperability (INTEROP 2016) at LREC 2016*, pages 5–9, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Godfrey, J. and Holliman, E. (1993). Switchboard-1 Release 2 LDC97S62. Web Download, Philadelphia: Linguistic Data Consortium `https://catalog.ldc.upenn.edu/ldc97s62`.

Goecks, J., Nekrutenko, A., Taylor, J., and Team, T. G. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.*, 8(11).

Replicating Speech Rate Convergence Experiments on the Switchboard Corpus

Simone Fuscone[1,2], Benoit Favre[2] and Laurent Prévot[2,3]
[1] Aix-Marseille Univ, CNRS, LPL, Aix-en-Provence, France
[2] Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
[3] Institut Universitaire de France, Paris, France

Abstract

Replicability of scientific studies grounded on language corpora requires a careful approach of each step from data selection and preprocessing up to significance testing. In this paper, we propose such a replication of a recent study based on a classic conversational corpus (Switchboard). The study (Cohen Priva et al., 2017) focuses on speech rate convergence between speakers in conversation. While the replication confirms the main result of the original study, it also shows interesting variations in the details. Moreover we take this opportunity to test further the study for its robustness with regard to data selection and preprocessing as well as to the underlying model of the variable (speech rate). The analysis also shows that another approach is necessary to consider the complex aspects of the speech rate in conversations. Pushing further a previous analysis is another benefit of replication in general: testing and strengthening the results of other teams and increasing validity and visibility of interesting studies and results.

Keywords: replication study, convergence, speech rate

1. Introduction

Convergence phenomena are well known in the speech science community: two speakers tend to co-adjust their speaking style in order to ease communication. In a conversation, we call each conversant as 'speaker' and his counterpart as 'interlocutor'. Behavior coordination between a speaker and his interlocutor has been shown to occur at various levels, like syntactic structures or referring expressions (Pickering and Garrod, 2004) and to accommodate to each other (Giles and Coupland, 1991). This study sought to replicate and expand the work of (Cohen Priva et al., 2017) which shows evidence about the phenomenon of accommodation. Their work focused on the speech rate convergence between speakers in the Switchboard corpus (Godfrey and Holliman, 1992), (Godfrey and Holliman, 1997). The goal of our study is at first show that it's possible to replicate the results of the work by Priva et al. following the same procedure and using the same statistical tools.

Replicability starts to receive a well-deserved attention from the Natural Language Processing (NLP) community. In language sciences and in particular in NLP, replicating a result may involve many detailed steps from the raw data to actual results. The replicability effort concerns therefore mainly the choices for data selection, pre-processing and the different steps in the analysis for which we try to follow the exact procedure of the replicated study. Interestingly, while the main lines and results of the replicated study are confirmed, particular results differ despite our care in not altering the original experimental setup. Moreover, based on our replication we can explore the robustness of the results by varying some of the parameters of the original study. We believe this is another interest in replicating a study.

The replication includes two parts: (i) one related to the effects of sex and age that affect the speech rate; (ii) and a second one which ensures the convergence of the speaker's speech rate to their baseline and the interlocutor's speech rate baseline. The second part will show further analysis performed on the corpus using the same model. At first we used different subsets of the main corpus changing the number of minimum conversations per speaker, then we tested another approach to compute the word expected duration and finally validated the model with a k-fold cross validation technique. In this last part, we also point out the necessity to have a different approach that could considers the temporal dynamic of speech rate, showing an example of the complex nature of the convergence.

The paper is organized as follows. After motivating the general interest for the research question (Section 2.), we present our replication (Section 3.) of the different experiments. Before concluding we propose some addition to the initial study in section 4. in particular with regard to data set selection, to the underlying model, and pointing out the issue of the dynamic of the speech rate

2. Motivation

Speech rate is one of the aspects in which convergence arises. In (Buller and Aune, 1992) some effects of the speech rate were shown in accommodation theory while (Manson et al., 2013) assess that convergence in speech rate predicts cooperation. As summarized in the work of (Cohen Priva et al., 2017), convergence during conversations could be attributed to the sex and age of the speakers (Hannah and Murachver, 1999; Kendall, 2009; Babel, 2012). Moreover, they recall that research has suggested women generally converge more than men (R.Bilous and M.Krauss, 1988; Gallois and Callan, 1988; Willemyns et al., 1997), though such results are often small and complex. Additional effects have been found on the interaction between speaker sex and interlocutor sex.[1] (Kendall, 2009) found that speech rates were more strongly affected by the interlocutor's sex than by the speaker's sex—both male and female speakers talked in a similar, slow rate when interviewed by a woman, and faster when the interviewer was a man. The aim of this work is to replicate and expand (Cohen Priva et al., 2017), in order to take into account speech

[1] For the sake of clarity, we will call the 'other speaker', the *'interlocutor'*.

rate as one of the aspects to study convergence in dyadic conversations.

The goal of the Cohen et al. study was at first to analyze the effects of age and sex on speech rate and then study convergence in terms speakers and interlocutors' baseline. As they showed a speaker may increase their usual speech rate (the baseline) in response to a fast-speaking interlocutor or *vice versa*. They also noted external factors could affect speech rate too. For example, controversial topics may incur faster speech rates as speakers get more involved. Previous work has measured convergence using third-party judgment (human judgment) (Namy et al., 2002; Goldinger, 1989), a comparison of the speech rate in the same conversation or comparing the speech rate with various baseline (Street, 1984; Levitan and Hirschberg, 2011; Pardo, 2006; Sanker, 2015). Priva et al. instead compared the speech rate of both conversants with their baseline performed in conversations with different speakers/interlocutors respectively.

This choice avoids the influence of inner factors in the conversations. As outlined in the previous work, in (Smith et al., 1975; Smith et al., 1980; Street, 1984) it is shown that people consider speakers with greater speech rate as more competent, so conversants could increase their speech rate to fit these impressions. Also, facial or body language expressions could affect the speech rate and the dynamic of the conversations. The use of the Switchboard dataset which is formed by telephone conversations with more than one conversation per speaker allows to smooth both these effects.

3. Replication Study

To ease the comparison with the work of Priva et al. we will use the same definitions. The speaker speech rate while speaking with the interlocutor I is indicated as S_I, while interlocutor speech rate with the speaker S is I_S. The speech rate baseline of the speaker in other conversations, with everyone except I is indicated as S_{-I}. Similarly I_{-S} is the speech rate baseline of the interlocutor while speaking with everyone except S.

The data used in the replication is the same of the paper by Priva et al., the Switchboard corpus (Godfrey and Holliman, 1992) in which participants took part in multiple telephonic conversations. There are 543 speakers in the corpus, with about 2400 conversations containing transcription for each dialogue, with conversants of mixed/same sex and ages. The speakers are strangers and each speaker was paired randomly by a computer operator with various other speakers; for each conversation a topic (from a list of 70 topics) was assigned randomly. In the pure replication stage were taken into account just conversations in which both conversants have at least one additional conversation with a different speaker, as in the original study. So after filtering the data by excluding speakers occurring in only one conversation we have 4788 sides of conversations and 479 speakers.

3.1. Speech Rate

In their work, Priva et al. computed *Pointwise speech rate* for an utterance as the ratio between *utterance duration* and *utterance expected duration*. The speakers speech rate was calculated as the mean of the log pointwise speech rates of all utterances having four or more words. Shorter utterances were removed because many of these are backchannels (Yngve, 1970), such as *'yeah'* or *'uhuh'*, which may exhibit specific phenomena with regard to speech rate. In addition, both the speakers and the interlocutors baseline speech rate were calculated using the mean speech rate of that caller in other conversations (S_I and I_S, respectively). Utterance expected duration was defined as the sum of the expected durations of all words in the utterance, excluding silences, filled pauses (*uh, um*) and *oh*. Utterance duration was defined as the time from the beginning of the first word in an utterance, which was not a silence or filled pause, until the end of the last word in that utterance, which was not a silence or filled pause, but including intermediate silences and filled pauses.

To calculate each words expected duration, Priva et al. used a linear regression model, in which the median duration of that word across the entire Switchboard corpus, the length of the utterance, and the distance to the end of the utterance (in words) are the predictors. Medians were used because the distribution of word durations is not symmetric. They included also the length of the utterance and the distance to the end of the utterance because it has been shown that both of these factors can affect rate of speech ((Jiahong Yuan, 1980; Quené, 2008; Jacewicz et al., 2009)). We find that the mean is 246 ms for both and the median 205 ms for actual, 208 ms for the expected.

3.2. Statistical Models

The model used for their analysis was a linear mixed regression model with the use of standardized speech rate as the predicted value. As specified the `lme4` library in `R` (Bates et al., 2014) was used to fit the models and provide t-values. The `lmerTest` package (Kuznetsova et al., 2014), which encapsulates `lme4`, was used to estimate degrees of freedom (*Satterthwaite approximation*) and calculate p-values. All numerical predictors were standardized. All models used the interlocutor, conversation, and topic identity as random intercepts. Study 1 also used the speaker as a random intercept. Binary predictors (speaker and interlocutor sex), were converted ("FEMALE") to 0 and ("MALE") to 1. Following the replication we used `Rs p.adjust` function to adjust p values for multiple comparisons using the FDR (false discovery rate) method.

3.3. Study 1: Sex and Age Effects on Speech Rate

This part of the work seeks to validate previous studies establishing that age and sex affect speech rate. In particular, younger speakers have been found to have faster rates than older speakers (Duchin and Mysak, 1987; Harnsberger et al., 2008; Horton et al., 2010), and male speakers slightly faster rates than female speakers (Jacewicz et al., 2009; Jiahong Yuan, 1980; Kendall, 2009). Sex, age, and their interaction were used as fixed effects. The models described used a random intercept for conversation.

Results As Priva et al., we find that older speakers are more likely to have a slower rate of speech (β = 0.2151,

	Estimate		Standard Error		FDR-adjusted p	
Variable	us	Priva et al.	us	Priva et al.	us	Priva et al.
Age	$\mathbf{0.2151}$	$\mathbf{0.2239}$	$\mathbf{0.0532}$	$\mathbf{0.0541}$	$\mathbf{1.2 \cdot 10^{-13}}$	$\mathbf{6.3 \cdot 10^{-5}}$
Speaker Sex	$\mathbf{-0.4089}$	$\mathbf{-0.3912}$	$\mathbf{0.0744}$	$\mathbf{0.0760}$	$\mathbf{2.5 \cdot 10^{-7}}$	$\mathbf{1.1 \cdot 10^{-6}}$
$Age \cdot Sex$	-0.0716	-0.0795	0.0748	0.0762	0.338	0.297

Table 1: Comparison of the results between our replication and the original study 1 from Cohen et al.

	Estimate		Standard Error		FDR-adjusted p	
Variable	us	Priva et al.	us	Priva et al.	us	Priva et al.
Speaker Baseline (SB)	$\mathbf{0.7777}$	$\mathbf{0.7940}$	$\mathbf{0.0929}$	$\mathbf{0.0090}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{2 \cdot 10^{-16}}$
Interlocutor's Baseline (IB)	$\mathbf{0.0464}$	$\mathbf{0.0540}$	$\mathbf{0.0094}$	$\mathbf{0.0190}$	$\mathbf{7 \cdot 10^{-6}}$	$\mathbf{0.034}$
Interlocutor's Age (IA)	$\mathbf{0.0231}$	$\mathbf{0.0249}$	$\mathbf{0.0089}$	$\mathbf{0.0100}$	$\mathbf{0.038}$	$\mathbf{0.043}$
$interlocutor's\ sex\ (IS)$	-0.0181	0.0099	0.0927	0.0230	0.134	0.844
$IB \cdot speaker\ Age$	0.0048	0.0025	0.0089	0.0090	0.720	0.844
$IB \cdot IA$	-0.0004	-0.0079	0.0082	0.0090	0.960	0.630
$IA \cdot speaker\ Age$	-0.2094	-0.0230	0.0092	0.0100	0.111	0.053
$IB \cdot speaker\ Sex$	-0.0075	0.0084	0.0092	0.0250	0.553	0.844
$IB \cdot SB$	-0.0173	-0.0176	0.0095	0.0100	0.183	0.162
$IB \cdot IS$	-0.0144	-0.0009	0.0093	0.0270	0.246	0.974
$IS \cdot Speaker\ Sex$	0.0022	-0.0676	0.0101	0.0270	0.945	0.430
$IB \cdot Speaker\ Age \cdot IA$	0.0064	0.0040	0.0078	0.0070	0.550	0.844
$IA \cdot Speaker\ Sex \cdot IS$	-0.0130	-0.0561	0.0091	0.0340	0.261	0.193

Table 2: Comparison of the results between our replication and the original study 2 from Priva et al

standard error (SE) = 0.0532, p < 10^{-5} , FDR-adjusted p < 10^{-6}). Male speakers are overall more likely to have a faster rate of speech (β = -0.4089, SE = 0.0744, p < 10^{-7} , FDRadjusted p < 10^{-6}). Age did not affect male and female speakers differently (β = -0.0716, SE = 0.0748, unadjusted p = 0.3389 , FDR-adjusted p > 0.05). These results summarized are shown in Table 1 and compared with the results of the work of Priva et al. As shown our work replicates the trend of the estimates of Priva et al., and the fact that both age and sex of speaker affect his speech rate.

3.4. Study 2: Converging to baseline

The second part of the original study attempted to determine to what extent speakers converge with their interlocutors baseline rate and to verify the influence of other features like sex and age on the convergence. The method is the same as explained in section 3.3.; moreover, were added several predictors. First two predictors for speech rate like the speakers baseline speech rate, estimated from their conversation with other interlocutors (S_I), and the interlocutors baseline speech rate, estimated from their conversations with others (I_S).
Other predictors are included, as described by Priva et al., to take into account that the identity of the speaker, both speaker and interlocutor properties like sex and age could affect the speech rate. The other predictors are:

- The age (standardized) of the interlocutor, as well as its interaction with the (standardized) age of the speaker: $Interlocutor\ age$; $Interlocutor\ age \cdot speakerage$
- The sex of the interlocutor, and its interaction with the sex of the speaker: $Interlocutor\ sex$; $Interlocutor\ sex \cdot speaker\ sex$
- Interactions between the interlocutors baseline speech rate and all other variables:
  - $Interlocutor\ Baseline \cdot Speaker\ Baseline$;
  - $Interlocutor\ Baseline \cdot Speaker\ Age$;
  - $Interlocutor\ Baseline \cdot Interlocutor\ Age$;
  - $Interlocutor\ Baseline \cdot Interlocutor\ Age \cdot speaker\ Age$;
  - $Interlocutor\ Baseline \cdot Speaker\ Sex$;
  - $Interlocutor\ Baseline \cdot Interlocutor\ Sex$;
  - $Interlocutor\ Baseline \cdot Interlocutor\ Sex \cdot Speaker\ Sex$.

Results As shown in table 2, our replication is in agreement with the results of Priva et al. Speakers baseline speech rate has the most significant effect on their own speech rate in a conversation (β = 0.7777, standard error (SE) = 0.0929, p < 10^{-16} , FDR-adjusted p < $2 \cdot 10^{-16}$). The interlocutors baseline rate has a smaller yet significant effect on speakers speech rate (β = 0.0464, standard error (SE) = 0.0094 , p < $8 \cdot 10^{-8}$, FDR-adjusted p < 0.05). The positive coefficient indicates convergence, when speaking with an interlocutor who speak slowly or quickly, the speakers speech rate changes in the same direction. Difference between the effect of speakers baseline rate and interlocutors baseline rate on speaker speech rate, suggests that speakers are more consistent than they are convergent, and rely much more on their own baseline.

	Speaker Baseline			Interlocutor's Baseline			Interlocutor's Age		
conv. per speaker	Estimate	SD	adj.p	Estimate	SD	adj.p	Estimate	SD	adj.p
2	$\mathbf{0.7777}$	$\mathbf{0.0093}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{0.0464}$	$\mathbf{0.0094}$	$\mathbf{7 \cdot 10^{-6}}$	$\mathbf{0.0231}$	$\mathbf{0.0089}$	$\mathbf{0.038}$
3	$\mathbf{0.7824}$	$\mathbf{0.0094}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{0.0588}$	$\mathbf{0.0192}$	$\mathbf{0.018}$	0.0217	0.0089	0.084
4	$\mathbf{0.7824}$	$\mathbf{0.0096}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{0.0589}$	$\mathbf{0.0194}$	$\mathbf{0.019}$	0.0205	0.0092	0.138
5	$\mathbf{0.7802}$	$\mathbf{0.0098}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{0.0589}$	$\mathbf{0.0197}$	$\mathbf{0.023}$	0.0206	0.0093	0.144
6	$\mathbf{0.7800}$	$\mathbf{0.0102}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{0.0652}$	$\mathbf{0.0200}$	$\mathbf{0.009}$	0.0220	0.0096	0.106

Table 3: Estimate, Standard deviation and adjusted p-value for the Speaker Baseline, Interlocutor's baseline and Interlocutor's age for different subsets of the Switchboard corpus. The subsets contain at least 2, 3, 4 ,5 and 6 conversations per speakers respectively.

Interlocutor age has a significant effect on speaker speech rate too (β = 0.0231, SE = 0.0089, p < 0.05, FDR-adjusted p < 0.05). The positive coefficient of this variable indicates that speakers are categorically slower while speaking with older speakers, regardless of the interlocutors baseline speech rate.
However, contrarily to the results of Priva et al. we don't find significance to assess that the combination of speakers and interlocutors sex affects speech rate.

4. Strenghting The Analysis

In this part we will show further analysis performed on the Switchboard corpus to test the model proposed by (Cohen Priva et al., 2017). More precisely, we extend the study in three directions: (i) using a subset of the corpus in order to include just speakers with more conversations; (ii) applying a different model to compute the word expected duration, and (iii) finally testing the model on different data subsets following a k-fold approach.

4.1. Taking a More Conservative Stance on Baseline Estimate

As said before, external factors could affect speech rate, like the topic of the conversation. Indeed, a speaker could vary his speech rate depending on how he is immersed into the discussion or according to the importance he gives to the topic. This may have an effect on the computation of the baseline leading to an overestimating or underestimating of the speech rate baseline. To smooth this effect we apply the same model to subsets of the Switchboard corpus considering just speakers who have at least 2, 3, 4, 5, 6 conversations, in order to have a greater number of conversations per speakers to compute S_I and I_S even if this implies to consider into the analysis a smaller number of total speakers; in this way we obtain 5 different datasets with respectively 479, 442, 406, 385, 357 different speakers and 4788, 4630, 4418, 4264 and 4018 'conversations sides'. The choice of using these datasets is also due to other factors, such as the internal state of the speaker. For example, emotion could affect the way to talk of a speaker and subsequently his speech rate. In previous studies, (Ververidis and Kotropoulos, 2006) compared the effect of the emotion to recognize them by the analysis of speaking using sverals database, while (Siegman and Boyle, 1993) outlined people who feel sadness can speak slow and soft. Using a greater numbers of conversations per speakers it could be possible to smooth this effects in the computation of the baseline. For study 2, We consider just predictors which were significant in the previous study and that still remain the only significant variables. Table 4 shows the magnitude of the estimates (for study 1) for each subsets. The magnitude of the effect of sex on speech rate increases with the number of conversations, while the effect of age decreases. Moreover, both variables preserve significance with an adjusted p-value that in the worst case (corresponding to the dataset with 6 conversations per speaker) is $p = 0.009$ for speaker age and $p \sim 10^{-8}$ for speaker sex. So, the trend of the estimates, still significant, suggests that considering a less quantity of data, lead to indicate the robustness of the model.
With regard to study 2 we consider just significant predictors. The results in table 3 shows that also in this case the magnitude of the speakers baseline, interlocutors baseline and of interlocutors age increase, but we note that the age loses significance as the number of minimum conversations increases. The speech rate so results to be affected mainly by the speaker baseline and by the interlocutor's baseline. Moreover, the fact that the interlocutor age doesn't seem to affect the convergence of speech rate, which implies the results can't be reproduced if we reduce the size of the dataset, recall the issue outlined by (Benjamin et al., 2017); in their work they suggest the possibility to reduce (for same fields of the scientific research) the threshold of the p-value, in order to help the reproducibility task of the results in the scientific community.

4.2. Variation on Expected Duration Computation

Recalling the definition of speech rate at a level of an utterance as the ratio between utterance duration and utterance expected duration, it's clear that the speech rate is influenced by the way of computing the expected duration of each individually word. Assuming that the duration of a word depends on the length of the utterance, the distance to the end and to the median duration of that word in the entire corpus, we fitted the expected duration using an artificial neural network regression with a one-hidden layer of 10 neurons and an adaptive learning method. The model is integrated by the use of the Scikit-Learn package in Python (Pedregosa et al., 2011). In this case we obtained that median of the expected word duration is $\sim$ 205 ms, as the median of the word duration in the corpus. Applying the same procedure of the previous paragraph, we obtained the

	Speaker Sex			Speaker Age			Sex·Age		
conv. per speaker	Estimate	SD	adj.p	Estimate	SD	adj.p	Estimate	SD	adj.p
2	$\mathbf{-0.4090}$	$\mathbf{0.0745}$	$\mathbf{2.6 \cdot 10^{-7}}$	$\mathbf{0.2151}$	$\mathbf{0.0532}$	$\mathbf{1.22 \cdot 10^{-4}}$	-0.0716	0.0748	0.338
3	$\mathbf{-0.4655}$	$\mathbf{0.0765}$	$\mathbf{1.04 \cdot 10^{-8}}$	$\mathbf{0.2083}$	$\mathbf{0.0544}$	$\mathbf{2.00 \cdot 10^{-4}}$	-0.0269	0.0770	0.726
4	$\mathbf{-0.4861}$	$\mathbf{0.0787}$	$\mathbf{6.42 \cdot 10^{-9}}$	$\mathbf{0.2055}$	$\mathbf{0.0564}$	$\mathbf{4.03 \cdot 10^{-4}}$	-0.0593	0.0794	0.455
5	$\mathbf{-0.4749}$	$\mathbf{0.0807}$	$\mathbf{3.49 \cdot 10^{-8}}$	$\mathbf{0.1854}$	$\mathbf{0.0585}$	$\mathbf{2.21 \cdot 10^{-3}}$	-0.0559	0.0822	0.496
6	$\mathbf{-0.4747}$	$\mathbf{0.0830}$	$\mathbf{9.26 \cdot 10^{-8}}$	$\mathbf{0.1634}$	$\mathbf{0.0602}$	$\mathbf{9.25 \cdot 10^{-3}}$	-0.0032	0.0847	0.968

Table 4: Estimate, Standard deviation and adjusted p-value for the Sex, age and $sex \cdot age$ for different subsets of the Switchboard corpus. The subsets contain at least 2, 3, 4 ,5 and 6 conversations per speakers respectively.

	Speaker Baseline			Interlocutor's Baseline			Interlocutor's Age		
conv. per speaker	Estimate	SD	adj.p	Estimate	SD	adj.p	Estimate	SD	adj.p
2	$\mathbf{0.7801}$	$\mathbf{0.0093}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{0.0548}$	$\mathbf{0.0192}$	$\mathbf{0.035}$	$\mathbf{0.0232}$	$\mathbf{0.0088}$	$\mathbf{0.048}$
3	$\mathbf{0.7868}$	$\mathbf{0.0094}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{0.0584}$	$\mathbf{0.0192}$	$\mathbf{0.019}$	0.0219	0.0089	0.078
4	$\mathbf{0.7847}$	$\mathbf{0.0096}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{0.0581}$	$\mathbf{0.0194}$	$\mathbf{0.022}$	0.0206	0.0091	0.137
5	$\mathbf{0.7822}$	$\mathbf{0.0097}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{0.0583}$	$\mathbf{0.0190}$	$\mathbf{0.024}$	0.0210	0.0093	0.140
6	$\mathbf{0.7970}$	$\mathbf{0.0100}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{0.0650}$	$\mathbf{0.0240}$	$\mathbf{0.009}$	0.0217	0.0095	0.093

Table 5: The table reports the results obtained using the method described in Section 4.2. to compute the expected word duration. Estimate, Standard deviation and adjusted p-value for the Speaker Baseline, Interlocutor's baseline and Interlocutor's age for different subsets with at least 2, 3, 4 ,5 and 6 conversations.

	estimate		SD		adj-p	
Variable	k-fold	previous	k-fold	previous	k-fold	previous
Speaker Baseline	$\mathbf{0.764 \pm 0.011}$	$\mathbf{0.778}$	$\mathbf{0.011 \pm 0.002}$	$\mathbf{0.009}$	$\mathbf{2 \cdot 10^{-16}}$	$\mathbf{2 \cdot 10^{-16}}$
Interlocutor Baseline	0.055 ± 0.007	0.046	0.001 ± 0.010	0.009	0.110 ± 0.071	$7 \cdot 10^{-6}$
Interlocutor's Age	0.016 ± 0.008	0.023	0.011 ± 0.001	0.009	0.130 ± 0.029	0.038

Table 6: Estimate, Standard deviation and adjusted p-value for the Speaker Baseline, Interlocutor's baseline and Interlocutor's age averaged on the 5 different subsets and compared with the value computed in Section 3.4.

.

results in table 5. The trend of the estimates and SD results similar to what founded in Section 4.1., reinforcing the hypothesis that both speaker baseline and interlocutor baseline affect the speech rate.

4.3. Validation of the Model

To validate the model described in the previous section, we apply a cross validation k-fold approach to determine if the results are still significant on a smaller independent dataset. We use k = 5, obtaining each subsets from the main corpus; differently from the section 4.1., we filter the data in order to create dataset with a size of conversations number corresponding to the 80% of the total length of the corpus, used in section 3.. In this context, each of data set contained 3830 'conversations sides' with the condition that each speaker has at least 2 conversations. We compare the results of study 2 3.4. with the results averaged on the subsets as expressed in Table 6. We found that even if Interlocutor's baseline and Interlocutor's Age (estimate and Standard deviation values) are consistent with the value of Section 3., they are not still significant. Vice-versa, the estimate for the Speaker Baseline emerges to be slightly lower compared to the previous result, but it still have significance. The non significance values cannot be attributed to the decrease of speakers in the datasets. Indeed, the minimum number of speakers result to be 452, that is about the 95% of the total number used in Section 3.. These difference of results could be attributed to the use of less conversations sides per speaker in the k-fold subsets (after the filtering processing), that reinforce the hypothesis to consider more than 2 conversations per speaker. These results suggest the fact that speech rate is mainly affected by the Speaker baseline also when both the size of the conversations and the number of speakers decrease.

4.4. Speech rate as dynamic variable

The replication study performed in this work, including the further analysis about the robustness of the model, used speech rate averaged on all the utterances per each conversation. Even if this approach captures general properties and behavior of the speakers and their interlocutors while conversing, it can't account for the dynamic of speech rate and how it evolves during time. In order to get a closer view to what speech rate variation looks like in conversation we realized a series of speech rate plots in actual conversation as shown in figure 1.

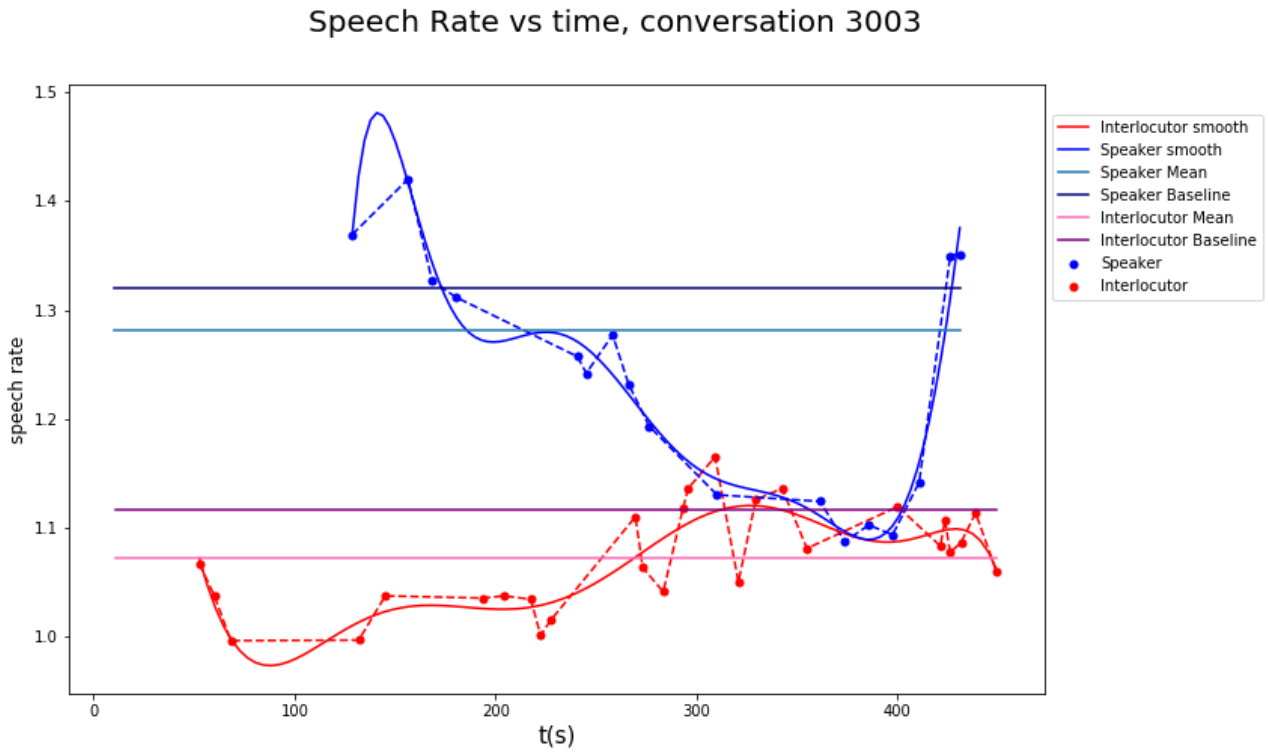

Figure 1: Blue shade (upper part) and red shade (bottom part) indicate respectively the speaker and interlocutor variables.

First of all, we should highlight that what does the replicated study (study 2) is to compare the base lines and the averages speech rates (all the straight lines). To show the variability and the complexity of the speech rate in a conversation we plot the speech rate for each utterance for both speaker and interlocutor. As first step we smooth the data using a moving average with a window n = 6. Then we apply a polynomial fit $p(x)$ of order $k = 8$ to the filtered data to obtain the trend of the speech rate as a smoothed function. As we can see, the difference between average speech rate of speaker and interlocutor (respectively in light blue and pink) is ~ 0.4. These averaged values are in according with the punctual speech rate (blue for speaker and red for interlocutor) at a level of the utterances for the first part of the conversation (up to 300 s) showing a huge difference between the conversants, but they hide that in the temporal interval $300-400\,s$ the difference is < 0.05. In the last part of the conversation speaker and interlocutor have a similar trend in speech rate and a model that uses the average speech rate can't take the temporal dynamic into account. Moreover, the average speech rate is sensitive to outliers. This issue could affect the analysis of the speech rate during conversation leading to an erroneous description of the conversants behavior. The importance to analyze the trend of speech rate that evolves during the conversation, points out the possibility of analyze speech rate with the use of new approaches that could study the dynamic of the conversation.

5. Conclusion

Our replication of (Cohen Priva et al., 2017) confirms that both speaker baseline and interlocutor baseline have effects on the speech rate, supporting the theory that speakers tend to convergence in speech rate as assesed in the work of (Cohen Priva et al., 2017). Although we test the robustness of their model, showing that only speaker baseline preserve significance in the test we performed.

More general, despite their key importance, replication studies in Language Sciences of the kind presented here have been too rare. However it is a crucial ingredient for making scientific results more reliable and more credible outside the community. It is important that this approach could be moved in other scientific fields to develop within Language Sciences. Moreover replicated studies are the best ground for extending previous work. We hope that the benefits exhibited in the paper can convince more NLP researchers to initiate replications and present them in dedicated papers.

Finally, the visual exploration of speech rate we presented allowed us to grasp the distance between the study we focused on, our replication and the actual complexity of the phenomena. It does not reduce the interest of the original study but reveals how we still need to understand about conversational dynamics.

6. Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No713750. Also, it has been carried out with the financial support of the Regional Council of Provence-Alpes-Côte dAzur and with the financial support of the A*MIDEX (n ANR- 11-IDEX-0001-02), funded by the Investissements d'Avenir project funded by the French Government, managed by the French National Research

Agency (ANR). Research also supported by grants ANR-16-CONV-0002 (ILCB), ANR-11-LABX-0036 (BLRI).

7. Bibliographical References

Babel, M. (2012). Evidence for phonetic and social selectivity in spontaneous phonetic imitation. *Journal of Phonetics*, 40(1):177 – 189.

Bates, D., Maechler, M., Bolker, B., Walker, S., et al. (2014). lme4: Linear mixed-effects models using eigen and s4. *R package version*, 1(7):1–23.

Benjamin, D. J., Berger, J. O., Johannesson, M., Nosek, B. A., Wagenmakers, E.-J., Berk, R., Bollen, K. A., Brembs, B., Brown, L., Camerer, C., et al. (2017). Redefine statistical significance. *Nature Human Behaviour*, page 1.

Buller, D. B. and Aune, R. K. (1992). The effects of speech rate similarity on compliance: Application of communication accommodation theory. *Western Journal of Communication*, 56(1):37–53.

Cohen Priva, U., Edelist, L., and Gleason, E. (2017). Converging to the baseline: Corpus evidence for convergence in speech rate to interlocutor's baseline. *The Journal of the Acoustical Society of America*, 141(5):2989–2996.

Duchin, S. W. and Mysak, E. D. (1987). Disfluency and rate characteristics of young adult, middle-aged, and older males. *Journal of communication disorders*, 20(3):245–257.

Gallois, C. and Callan, V. J. (1988). Communication accommodation and the prototypical speaker: Predicting evaluations of status and solidarity. *Language and Communication*, 8(3):271 – 283. Special Issue Communicative Accomodation: Recent Developments.

Giles, H. and Coupland, N. (1991). *Language: Contexts and consequences.* Thomson Brooks/Cole Publishing Co.

John J. Godfrey and Edward Holliman. (1992). *Switchboard-1 Release 2*. distributed via LDC, Switchboard, 2.0, ISLRN LDC97S62.

Goldinger, S. D. (1989). Echoes of echoes? an episodic theory of lexical access. *Psychological Review*, 105(2):251–279.

Hannah, A. and Murachver, T. (1999). Gender and conversational style as predictors of conversational behavior. *Journal of Language and Social Psychology*, 18(2):153–174.

Harnsberger, J. D., Shrivastav, R., Brown, W., Rothman, H., and Hollien, H. (2008). Speaking rate and fundamental frequency as speech cues to perceived age. *Journal of voice*, 22(1):58–69.

Horton, W. S., Spieler, D. H., and Shriberg, E. (2010). A corpus analysis of patterns of age-related change in conversational speech. *Psychology and aging*, 25(3):708.

Jacewicz, E., Fox, R. A., O'Neill, C., and Salmons, J. (2009). Articulation rate across dialect, age, and gender. *Language Variation and Change*, 21(2):233256.

Jiahong Yuan, Mark Liberman, C. C. (1980). Towards an integrated understanding of speaking rate in conversation. *Proceedings of Interspeech, Pittspurgh*, pages 541–544.

Kendall, T. (2009). Speech rate, pause, and linguistic variation: An examination through the sociolinguistic archive and analysis project. *Phd Thesis, Duke University*.

Kuznetsova, A., Bruun Brockhoff, P., and Haubo Bojesen Christensen, R. (2014). lmertest: tests for random and fixed effects for linear mixed effects models. *See https://CRAN. R-project. org/package= lmerTest*.

Levitan, R. and Hirschberg, J. (2011). Measuring acoustic-prosodic entrainment with respect to multiple levels and dimensions.

Manson, J. H., Bryant, G. A., Gervais, M. M., and Kline, M. A. (2013). Convergence of speech rate in conversation predicts cooperation. *Evolution and Human Behavior*, 34(6):419 – 426.

Namy, L. L., Nygaard, L. C., and Sauerteig, D. (2002). Gender differences in vocal accommodation:: The role of perception. *Journal of Language and Social Psychology*, 21(4):422–432.

Pardo, J. (2006). On phonetic convergence during conversational interaction. *The Journal of the Acoustical Society of America*, 119(2382).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Pickering, M. J. and Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and brain sciences*, 27(2):169–190.

Quené, H. (2008). Multilevel modeling of between-speaker and within-speaker variation in spontaneous speech tempo. *The Journal of the Acoustical Society of America*, 1104(123).

R.Bilous, F. and M.Krauss, R. (1988). Dominance and accommodation in the conversational behaviours of same- and mixed-gender dyads. *Language and Communication*, 8(3):183–194.

Sanker, C. (2015). Comparison of phonetic convergence in multiple measures. *in Cornell Working Papers in Phonetics and Phonology*, page 6075.

Siegman, A. W. and Boyle, S. (1993). Voices of fear and anxiety and sadness and depression: the effects of speech rate and loudness on fear and anxiety and sadness and depression. *Journal of Abnormal Psychology*, 102(3):430.

Smith, B. L., Brown, B. L., Strong, W. J., and Rencher, A. C. (1975). Effects of speech rate on personality perception. *Language and Speech*, 18(2):145–152.

Smith, B. L., Brown, B. L., Strong, W. J., and Rencher, A. C. (1980). Effects of speech rate on personality attributions and competency evaluations.

Street, R. L. (1984). Speech convergence and speech evaluation in fact-finding iinterviews. *Human Communication Research*, 11(2):139–169.

Ververidis, D. and Kotropoulos, C. (2006). Emotional speech recognition: Resources, features, and methods. *Speech communication*, 48(9):1162–1181.

Willemyns, M., Gallois, C., Callan, V. J., and Pittam, J. (1997). Accent accommodation in the job interview: Im-

pact of interviewer accent and gender. *Journal of Language and Social Psychology*, 16(1):3–22.

Yngve, V. H. (1970). On getting a word in edgewise. In *Chicago Linguistics Society, 6th Meeting*, pages 567–578.

8. Language Resource References

John J. Godfrey and Edward Holliman. (1992). *Switchboard-1 Release 2*. distributed via LDC, Switchboard, 2.0, ISLRN LDC97S62.

John J. Godfrey and Edward Holliman. (1997). *Switchboard-1 Telephone Speech Corpus*. distributed via LDC, Switchboard, 2.0, ISLRN 988-076-156-109-5.