Dealing with Abbreviations in the Slovenian Biographical Lexicon

Angel Daza[1] **Antske Fokkens**[1] **Tomaž Erjavec**[2]
[1]CLTL, Vrije Universiteit Amsterdam
[2]Department of Knowledge Technologies, Jožef Stefan Institute
{j.a.dazaarevalo,antske.fokkens}@vu.nl, tomaz.erjavec@ijs.si

Abstract

Abbreviations present a significant challenge for NLP systems because they cause tokenization and out-of-vocabulary errors. They can also make the text less readable, especially in reference printed books, where they are extensively used. Abbreviations are especially problematic in low-resource settings, where systems are less robust to begin with. In this paper, we propose a new method for addressing the problems caused by a high density of domain-specific abbreviations in a text. We apply this method to the case of a Slovenian biographical lexicon and evaluate it on a newly developed gold-standard dataset of 51 Slovenian biographies. Our abbreviation identification method performs significantly better than commonly used ad-hoc solutions, especially at identifying unseen abbreviations. We also propose and present the results of a method for expanding the identified abbreviations in context.

1 Introduction

Abbreviations such as "b." for "born", or "gr." for "graduated" are a common issue when dealing with digitized texts which use a large number of them for space-saving reasons. They are also a known problem when processing technical documents (Park and Byrd, 2001) and biomedical texts (Jin et al., 2019). In this paper, we examine the case of biographical dictionaries, i.e. collections of biographies that have been digitized, and, in particular, the Slovenian Biographical Lexicon.

To automatically extract facts from biographical texts, Digital Humanities researchers normally rely on out-of-the-box NLP tools such as Stanza (Qi et al., 2020) or SpaCy.[1] These tools are often adequate for identifying sentences which are then used as input for higher-level downstream tasks, for manual inspection, and for visualization purposes. However, out-of-the-box tools are designed to work for the broadest possible text domains and cover the most common cases. This impacts performance significantly when dealing with domain-specific data, such as entries of biographical dictionaries, and more so when they contain a lot of abbreviations. The problem is even more pronounced when dealing with a relatively lower resource language, such as Slovenian. The performance bottleneck occurs already at the first step, i.e. tokenization: in order to perform good tokenization in domain-specific texts, we need to have a reliable method for identifying abbreviations such that the tokenizer does not split them wrongly, generating faulty tokens and incomplete sentences. In this paper we:

- Quantify the effect that abbreviations have on a downstream task such as NER on Slovenian biographical texts.
- Propose a method for abbreviation identification, apply it to raw texts and compare it to straightforward baselines.
- Analyze the feasibility of using contextually dependent word embeddings, in particular, the SloBERTa (Ulčar and Robnik-Šikonja, 2021) language model, to automatically expand abbreviations in text and improve readability.
- Evaluate the performance of our methods on a new human-curated dataset with, inter alia, gold tokens, sentences, named entities, and expanded abbreviations.[2]

2 Related Work

Specific work addressing abbreviations is scarce. We think this is due to the fact that it is addressed as a preprocessing step with tailored solutions for each specific use case, involving regular expressions, or corpus-specific rules (Bollmann et al., 2011). A few papers try to construct methods for a general solution to this problem. For instance,

[1] https://spacy.io/

[2] https://github.com/angel-daza/abbreviation-detector

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8715–8720
December 7-11, 2022 ©2022 Association for Computational Linguistics

Park and Byrd (2001) propose a pipeline system to induce acronyms, where every sequence of characters (separated by spaces) is considered a *candidate abbreviation* if it satisfies certain conditions. Żelasko (2018) proposes a more advanced approach based on an LSTM classifier that uses morphosyntactic information about a sentence to directly infer the correct expansion of an abbreviation in Polish. Direct work on abbreviations also exists in the biomedical domain, including detection and disambiguation (Stevenson et al., 2009) as well as abbreviation expansion (Jin et al., 2019). Finally, Gorman et al. (2021) have recently developed an English dataset to explore abbreviation expansion methods taking into account the context.

Another common way to address the difficulty of domain-specific texts is *text normalization*, which is the task of *translating* a domain-specific text into more standard form (this can be at different levels such as lexical or morphological) that is easier to process by general purpose NLP tools. This approach is common when dealing with user generated text and social media (Pennell and Liu, 2011; Baldwin et al., 2015; van der Goot, 2019), and also historical texts (Scherrer and Erjavec, 2013; Ljubesic et al., 2016; Bollmann, 2019).

One drawback of text normalization is that it is frequently implemented using an Encoder-Decoder approach (Robertson and Goldwater, 2018; Bollmann et al., 2019), which requires a big-enough parallel corpus to be trained and obtain good-quality results. Another drawback is the fact that it *generates* a new standard sentence, which is not a desired side effect if we want to preserve word by word the original biographical text. In contrast to the normalization task, we are interested in preserving the original text and only identifying (and perhaps expanding) the abbreviations that are problematic for the NLP tools.

3 Dataset

The Slovenian Biographical Lexicon (SBL) was published in 15 volumes (1925–1991) and contains 5,047 biographies (Ogrin et al., 2013). For the experiments described in this paper, we have created the dataset SBL-51abbr (Erjavec et al., 2022), which consists of 51 randomly selected entries from SBL[3]. The text of each entry is manually tokenized and sentence segmented, marked with named entities, and lemmatized words. It has also been automatically annotated with Universal Dependencies PoS tags, morphological features and dependency parses using CLASSLA (Ljubešić and Dobrovoljc, 2019),[4] a fork of the Stanford Stanza pipeline (Qi et al., 2020),[5] which is the state-of-the-art tool for annotating Slovenian. Crucially for the envisaged use of the corpus, the abbreviations in the corpus have been manually expanded so that the expansions are in the correct inflected form. The curated dataset consists of 655 sentences (see Table 1). It is available in the canonical TEI encoding, and derived plain text and CoNLL-U files. The plain-text file has abbreviations and their expansions marked up with [[...]]((...)) respectively. There are two CoNLL-U files, one with the text stream with abbreviations, and one with the text stream with expansions. Note that only the one with expansions has syntactic parses. Both CoNLL-U files have the expansions / abbreviations and named entities marked up in IOB format in the last column.

We use this dataset as a gold standard to test the performance of our proposed methods. We randomly split the available data into three portions: 70% for training, 10% for development and 20% for testing.

Split	Sents	Abbrs	Unique	Unseen
Train	458	1385	399	0
Dev	66	236	130	33
Test	131	420	181	70
Σ	655	2041	710	

Table 1: Abbreviation statistics on the SBL-51abbr dataset. We count the total number of abbreviations, the unique types and the number of unseen abbreviations in the dev and test splits.

4 Impact of Abbreviations

We first quantify the impact that the high number of abbreviations in the SBL-51abbr corpus has when processing the raw texts with CLASSLA (Ljubešić and Dobrovoljc, 2019) and performing NER. We compare the performance on the original texts (with all abbreviations) with a second scenario where abbreviations were substituted with their gold expansions. Table 2 shows the performance per class when processing the original version (the first row of numbers per label), and right below is the perfor-

[3] http://hdl.handle.net/11356/1588

[4] https://pypi.org/project/classla/

[5] https://stanfordnlp.github.io/stanza/

mance when processing the same texts but without any abbreviation. There is a significant boost all across the board for the fully expanded texts, resulting on over 30 F1 points of improvement on the macro average measure. This shows that having effective methods for abbreviation identification and expansion can be beneficial.

Label	P	R	F1
PER	68.75	22.45	33.85
	76.80	65.31	70.59
DERIV-PER	50.00	4.76	8.70
	92.86	61.90	74.29
LOC	85.29	39.73	54.21
	82.32	92.47	87.10
MISC	65.38	12.41	20.86
	56.14	23.36	32.99
ORG	23.53	14.81	18.18
	22.06	55.56	31.58
macro_avg	58.59	18.83	27.16
	66.03	59.72	59.31

Table 2: NER scores when applied to the original sentences with abbreviations (upper rows) vs sentences with all abbreviations expanded (lower rows).

5 Dealing with Abbreviations

5.1 Baselines

Dictionary-based: to bypass tokenizer-specific noise, we split every document by spaces to obtain a list of *dirty tokens*.[6] For each token, we first clean it, meaning we remove all special characters except full stops. If the *clean token* does not end with a full stop we skip it, otherwise we strip the full stop and check if the entry exists in a large dictionary. We use two dictionaries: the Hunspell dictionary,[7] a popular tool used for spelling correction, and GigaFida 2.0 (Krek et al., 2020), a big reference corpus of standard Slovene. Because dictionaries only contain entries for complete words, we consider the token an abbreviation if no entry exists.

Corpus-based: we tokenize the training corpus using CLASSLA and compute the frequency for all token unigrams t_1 and bigrams (t_1, t_2) in the corpus. If a bigram contains a full stop as a second component or if a unigram has a full stop as its last character,[8] then we increase the count of t_1 in $\mathcal{A}$ otherwise we increase the count of t_1 in $\mathcal{B}$. We take all t_1s that appear in both lists and calculate their probability to be an abbreviation as:

$$P(t_1 = abbr) = \frac{freq(t_1 \epsilon \mathcal{A})}{freq(t_1 \epsilon \mathcal{A}) + freq(t_1 \epsilon \mathcal{B})} \quad (1)$$

If the probability $P(t_1 = abbr) \geq 0.8$ then t_1 is considered to be an abbreviation, otherwise we skip it. This method will, of course, carry over some of the tokenization mistakes. The reasoning behind this baseline is to capture the number of times a given token appears before a full stop compared to the total times it appears in the corpus. If it is the case that most of the occurrences of such a token are immediately followed by a full stop, then it is most likely an (unrecognized) abbreviation.

5.2 Abbreviation Classifier

We propose an automatic method for identifying abbreviations by fine-tuning a classifier on top of the SLoBERTa language model (Ulčar and Robnik-Šikonja, 2021). We again obtain the sequence of tokens by splitting the raw texts by spaces. We treat each one of the *dirty tokens* as a separate input sequence to SloBERTa. We train the classifier using the gold abbreviation labels to predict if a token is an abbreviation or not (Figure 1, bottom).

5.3 Abbreviation Expansion

Once we have our text with abbreviation candidates identified, for each candidate, we take the full sentence it appears in, mask it and let SloBERTa predict the masked token. We take SloBERTa's prediction to be a valid expansion if one of the top 5 predicted candidates starts with the same letter as the masked abbreviation, otherwise we leave the original abbreviation. This way we are substituting each candidate *in-context* and thus approaching the optimal scenario with the fully expanded sentences where NER performed much better (see Table 2). A visualization of both the identification and expansion steps of our method is given in Figure 1.

[6] We call them *dirty* because they will have punctuation attached to them. For example "Hello world!" will be *tokenized* as ['Hello', 'world!'] instead of ['Hello', 'world', '!'] which would be the optimal tokenization.

[7] `https://github.com/hunspell/hunspell`

[8] We test for this case because, if the tokenizer rightly recognized the abbreviation, then the full stop will still be attached to it

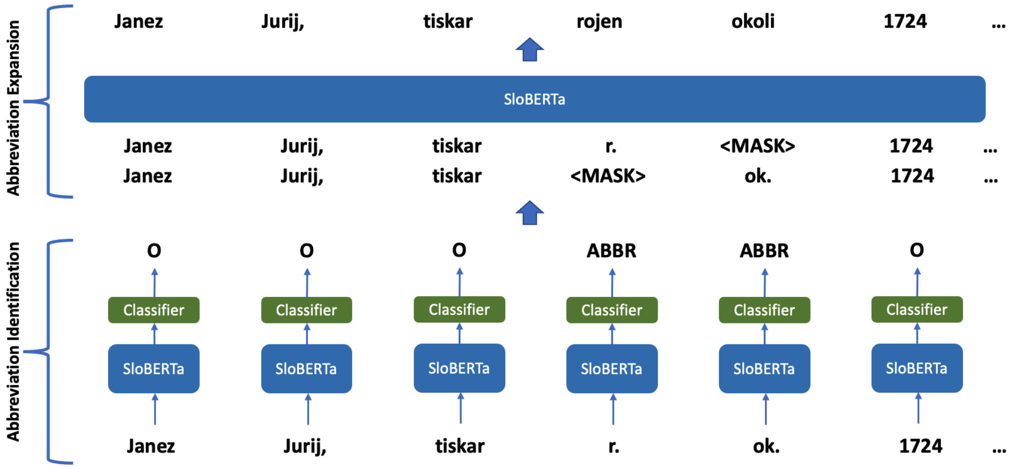

Figure 1: We first train a classifier for identifying abbreviations (SloBERTa+Neural linear layer) and then match each of the identified candidates to use SloBERTa as a predictor for that slot. We only keep the suggested expansion if it meets the requirements.

6 Results and Evaluation

6.1 Abbreviation Identification Baselines

We first present the results on the test set obtained by our proposed baselines in Table 3. These baselines represent common pre-processing approaches to dealing with abbreviations. We can see that the two dictionary versions suffer from low recall, especially the GigaFida dictionary with only 20%. This behavior is expected since we are dealing with a domain-specific (and partially historical) text. The second baseline behaves much better and achieves an 85.34 F1 score. The Bigrams+Dict version mixes both approaches, which improves the coverage of identified abbreviations, but unfortunately lowers the high precision of the bigram approach.

Baseline	P	R	F1
GigaFida Dict	89.36	20.00	32.68
Hunspell Dict	80.81	71.19	75.70
Corpus Bigrams	95.85	76.90	85.34
Bigrams+Dict	73.27	95.95	83.09

Table 3: Abbreviation identification baseline results on the test set. They show trade-offs between precision and recall. The bigrams method performs the best.

6.2 Abbreviation Classifier

The baselines show a trade-off between good precision or good recall. In contrast, Table 4 shows that our SloBERTa method significantly increases the recall without hampering precision. We fine-tuned SloBERTa[9] for 5 epochs and pick the model that performs best on the development set. We present the mean of 5 experiments with different random seeds together with the standard deviation. The results demonstrate that this is a stable approach for identifying abbreviations in text.

Split	P	R	F1
Dev	$95.91^{\pm 0.75}$	$97.91^{\pm 1.9}$	$96.89^{\pm 0.9}$
Test	$93.94^{\pm 1.5}$	$98.10^{\pm 2.0}$	$95.97^{\pm 1.3}$

Table 4: Abbreviation identification results with our SloBERTa binary classifier. Results on test are 10 points above the best baseline.

6.3 Abbreviation Expansion

We measure the success of our abbreviation expansion method by re-running the NER tagger on the sentences with the expanded abbreviations as predicted by SloBERTa (see Table 5). From the 420 abbreviations in the test set, 154 where expanded following our heuristic and the rest of abbreviations in the sentences were left untouched. We can compare these scores directly with our analysis from Table 2 and see that even though our method for expanding is quite basic, it already gets us closer to the ceiling scores (where all gold expansions were substituted). Important gains can be seen in all categories and the macro average score reached with the predicted expansions is 49.64 F1 which is 22 points above the 27.16 F1 obtained originally.

[9] We used the default settings from HuggingFace https://huggingface.co/EMBEDDIA/sloberta

Label	P	R	F1
PER	40.54	67.67	50.70
DERIV-PER	78.57	55.00	64.71
LOC	72.33	82.73	77.18
MISC	34.34	27.64	30.63
ORG	17.14	46.15	25.00
macro_avg	48.59	55.84	49.64

Table 5: The NER results on the test set after applying SloBERTa-based expansions show consistent improvements compared to the original sentences (cf. Table 2)

7 Conclusions

In this paper we focused on the task of Named Entity Recognition to quantify the impact of abbreviations in a text by comparing the performance of the CLASSLA NER tagger on the same sentences with and without abbreviations. We presented a gold-standard dataset consisting of 51 biographies in Slovenian (a limited-resource language) in a specialized text domain. We also presented a method for automatically identifying abbreviations and expanding them without the need for a tokenizer. The biggest advantage of our method is that it can be applied out-of-the-box for any language which has a large language model available and does not need ad-hoc training data or large fixed dictionaries. Our abbreviation identification classifier obtains better precision and better recall when compared to other straightforward approaches to identify abbreviations.

Finally, we presented a method that uses a pretrained language model to predict plausible expansions for the identified abbreviations. We notice that our method is still simple but already achieves better results than directly processing the original sentences with abbreviations. In future work we aim to explore more sophisticated methods for abbreviation expansion that allow us to further improve the readability of texts. We find our results encouraging for researchers working with limited domains who may find a similar approach helpful for improving performance in other tasks.

Limitations

The results presented in this paper have been evaluated on the specific use case of the Slovenian Biographical Lexicon. When considering to apply this approach to other use cases, the following limitations should be taken into account:

Good Use Case. The SBL is a good use case in the sense that this is a domain with a high density of abbreviations and thus a relatively high number of positive class examples. This means that the relatively small dataset was comparatively rich (i.e. another domain may require more data) and the potential of improving results is relatively high (i.e. identifying abbreviations may have less impact on downstream tasks in other domains).

Language Model Required. A large language model is needed for this approach and may not be available for many low or even medium resource languages. This is unfortunate, because this research aims to support relatively low resource languages that must rely on standard tools, because there are limited resources for creating new data sets and models.

Expanding Abbreviations Remains Largely Unsolved. The results for expanding abbreviations are still meagre. Even though the current approach is simple, it may already represent an upperbound due to the productive character of abbreviations in this domain, the rich inflection of Slovenian, and the considerable effort required to obtain more training data.

Acknowledgements

This work was partially supported by the EU Horizon 2020 project InTaVia: In/Tangible European Heritage - Visual Analysis, Curation and Communication (http://intavia.eu) under grant agreement No. 101004825.

References

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135, Beijing, China. Association for Computational Linguistics.

Marcel Bollmann. 2019. A large-scale comparison of historical text normalization systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3885–3898, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcel Bollmann, Natalia Korchagina, and Anders Søgaard. 2019. Few-shot and zero-shot learning for historical text normalization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 104–114, Hong Kong, China. Association for Computational Linguistics.

Marcel Bollmann, Florian Petran, and Stefanie Dipper. 2011. Rule-based normalization of historical texts. In *Proceedings of the Workshop on Language Technologies for Digital Humanities and Cultural Heritage*, pages 34–42, Hissar, Bulgaria. Association for Computational Linguistics.

Tomaž Erjavec, Petra Vide Ogrin, Jakob Lenardič, Mojca Mlinar Strgar, and Simona Frankl. 2022. Annotated sample of the slovenian biographical lexicon SBL-51abbr 1.0. Slovenian language resource repository CLARIN.SI.

Kyle Gorman, Christo Kirov, Brian Roark, and Richard Sproat. 2021. Structured abbreviation expansion in context. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 995–1005, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qiao Jin, Jinling Liu, and Xinghua Lu. 2019. Deep contextualized biomedical abbreviation expansion. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 88–96, Florence, Italy. Association for Computational Linguistics.

Simon Krek, Špela Arhar Holdt, Tomaž Erjavec, Jaka Čibej, Andraz Repar, Polona Gantar, Nikola Ljubešić, Iztok Kosem, and Kaja Dobrovoljc. 2020. Gigafida 2.0: The reference corpus of written standard Slovene. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3340–3345, Marseille, France. European Language Resources Association.

Nikola Ljubešić and Kaja Dobrovoljc. 2019. What does Neural Bring? Analysing Improvements in Morphosyntactic Annotation and Lemmatisation of Slovenian, Croatian and Serbian. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 29–34, Florence, Italy. Association for Computational Linguistics.

Nikola Ljubesic, Katja Zupan, Darja Fišer, and Tomaž Erjavec. 2016. Normalising slovene data: historical texts vs. user-generated content. In *KONVENS*.

Petra Vide Ogrin, Simona Frankl, Mojca Mlinar Strgar, Izidor Cankar, Tomaž Erjavec, and Joahim Dokler, editors. 2013. *Slovenski biografski leksikon, elektronski vir (Slovenian Biographical Lexicon, Digital Edition)*. Slovenian Academy of Sciences and Arts. `http://www.slovenska-biografija.si/kolofon/sbl/`.

Youngja Park and Roy J. Byrd. 2001. Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Deana Pennell and Yang Liu. 2011. A character-level machine translation approach for normalization of SMS abbreviations. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 974–982, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Alexander Robertson and Sharon Goldwater. 2018. Evaluating historical text normalization systems: How well do they generalize? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 720–725, New Orleans, Louisiana. Association for Computational Linguistics.

Yves Scherrer and Tomaž Erjavec. 2013. Modernizing historical Slovene words with character-based SMT. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 58–62, Sofia, Bulgaria. Association for Computational Linguistics.

Mark Stevenson, Yikun Guo, Abdulaziz Alamri, and Robert Gaizauskas. 2009. Disambiguation of biomedical abbreviations. In *Proceedings of the BioNLP 2009 Workshop*, pages 71–79, Boulder, Colorado. Association for Computational Linguistics.

Matej Ulčar and Marko Robnik-Šikonja. 2021. *Slovenian RoBERTa contextual embeddings model: SloBERTa 2.0*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/11356/1397`.

Rob van der Goot. 2019. MoNoise: A multi-lingual and easy-to-use lexical normalization tool. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 201–206, Florence, Italy. Association for Computational Linguistics.

Piotr Żelasko. 2018. Expanding abbreviations in a strongly inflected language: Are morphosyntactic tags sufficient? In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).