LEMMATIZING SERBIAN AND CROATIAN VIA STRING EDIT PREDICTION

Lenka BAJČETIĆ, [1,2] Vuk BATANOVIĆ, [2] Tanja SAMARDŽIĆ[3]
[1]School of Electrical Engineering, University of Belgrade
[2]Innovation Center of the School of Electrical Engineering, University of Belgrade
[3]University of Zurich

In this paper, we examine the effectiveness of lemmatizing texts in Serbian and Croatian using a pre-trained large language model fine-tuned on the task of string edit prediction. We define lemmatization as a tagging task, where each word-lemma transformation is represented as a string edit tag which encodes the necessary prefix and suffix alterations. Our approach is verified using the BERTić large language model and leads to improved results on the standard Serbian SETimes.SR and the standard hr500k Croatian dataset, as well as on ReLDI-NormTagNER-sr and ReLDI-NormTagNER-hr datasets. Its additional advantage is that it does not rely on any lexical databases, making it easily applicable to different text domains and language variants.

Keywords: Lemmatization, BERTić, SETimes.SR, hr500k, ReLDI-NormTagNER-sr, ReLDI-NormTagNER-hr

1 INTRODUCTION

Lemmatization entails assigning to each word form its base form (e.g., 'write' $\rightarrow$ 'write', 'writes' $\rightarrow$ 'write', 'wrote' $\rightarrow$ 'write', 'written' $\rightarrow$ 'write'). It used to be a crucial task in linguistic analysis and text processing, especially for highly inflected languages like Serbian and Croatian, but its role is changing in modern approaches.

The importance of lemmatization for NLP tasks when dealing with morphologically rich languages is tested by (Kutuzov & Kuzmenko, 2019), who "critically evaluate the widespread assumption that deep learning NLP models do not require lemmatized input". They conclude that the decisions about text pre-processing before training language models should consider the linguistic nature of the language in question. As a matter of fact, lemmatization may

not be necessary for English, but using lemmatized training and testing data for Russian yielded small but consistent improvements for word sense disambiguation. When it comes to Serbian, a positive impact of text lemmatization on model performances has been shown in models for sentiment analysis (Batanović & Nikolić, 2017) and semantic similarity (Batanović et al., 2018).

In morphologically rich languages, diverse sets of grammatical information are encoded within each word using inflections. Both Serbian and Croatian have seven grammatical cases, three genders, and two grammatical numbers and this information is represented through a wide variety of inflections, usually in the suffix form.

In highly inflected languages such as these, it is typical to tackle the problem of lemmatization using inflectional lexicons. These lexicons can easily become extremely large since the high number of different inflectional variants of many words dramatically increases the vocabulary size. Even if semi-automatic techniques are employed in their construction, the process of creating inflectional dictionaries is costly and time-consuming. Furthermore, the resulting lexicons are inherently limited in size and scope, especially when it comes to particular or niche domains. This leads to the issue of out-of-vocabulary words which cannot be properly processed in this approach.

An alternative to using inflectional dictionaries is to redefine lemmatization as a task of predicting sets of string edits. The generic transformation from a word to a lemma is done in four steps: 1) remove a suffix of length N_s; 2) add a new lemma suffix, L_s; 3) remove a prefix of length N_p ; 4) add a new lemma prefix, L_p. In the end, the tuple [N_s; L_s; N_p; L_p] defines the word-to-lemma transformation. For example, the tuple of necessary string edits for the word 'učio' to get the lemma 'učiti' is [1, 'ti', 0, 0]. In this way, lemmatization can be understood as a sequence labeling task where each token's tag is actually the tuple which represents the set of necessary string edits. This approach is inherently more easily applicable to different text domains and language variants, as it does not rely on any lexical databases.

This technique was proven to work well both on Serbian (Gesmundo & Samardžić, 2012b) and a set of eight different languages (Gesmundo & Samardžić, 2012a), but all previous experiments were performed without deep and transfer learning, and relied on hand-crafted features instead of embed-

dings. In this paper, we examine the proposed lemmatization method when used in conjunction with a modern large language model. Specifically, we fine-tune a pre-trained language model for string edit sequence prediction, with each token's string edit tag being formulated based on the set of edits necessary to transform it into its lemma. For this purpose we rely on BERTić (Ljubešić & Lauc, 2021), a large language model based on the ELECTRA architecture, which was trained on more than 8 billion tokens of text in Bosnian (800 million), Croatian (5.5 billion), Montenegrin (80 million) and Serbian (2 billion).

2 RELATED WORK

The history of the task of lemmatization includes many approaches, from applying hand-written set of rules (Koskenniemi, 1984; Plisson et al., 2004) to general character-level transducers, which learn the lemmatization rules from example pairs (word form, lemma) (Dreyer et al., 2008; Eger et al., 2016; Nicolai & Kondrak, 2016). With the introduction of deep learning, character-level transducers were redefined as a case of sequence-to-sequence models and the task was solved with bi-LSTM encoder-decoder networks (Bergmanis & Goldwater, 2018; Kondratyuk et al., 2018).

Character-level transducers can be too expressive leading to over-generalization and other avoidable mistakes. A solution to this is to learn a set of edits (edit scripts or edit trees) as a single label (Chrupala, 2006; Chrupala et al., 2008; Gesmundo & Samardžić, 2012b), which makes the task of lemmatization similar to part-of-speech tagging (POS) or named entity recognition (NER). In this way, one can incorporate the information about the context in predicting lemmas, which can help avoid mistakes caused by ambiguity. Lemmatization is often solved jointly with POS or morphosyntactic tagging due to their inter-dependence (Garabík & Mitana, 2022).

Although these general solutions are elegant and reusable, their performance can be limited by irregularities, which is why it is common to devise additional filters with dictionary look-up (Jursic et al., 2010). For example, the HuSpaCy model for Hungarian contains a hybrid lemmatizer utilizing both a neural model, dictionaries and hand-crafted rules (Berkecz et al., 2023).

The CLASSLA-Stanza package is a pipeline for automatic linguistic annotation of the South Slavic languages, including Croatian and Serbian (Terčon & Ljubešić, 2023). The lemmatizer model is trained after morphosyntactic tagging is already performed, so it is utilizing both the tokens and the morphosyntactic tags, and it relies on an inflectional lexicon which serves as an additional controlling element during lemmatization. The Croatian model is trained using the hrLex 1.3 inflectional lexicon (Ljubešić, 2019a), while the Serbian model relies on srLex 1.3 inflectional lexicon (Ljubešić, 2019b).

3 METHODOLOGY

The goal of this work is to see whether defining lemmatization as a string edit prediction task will prove to be a suitable framing for a large language model to learn, thereby avoiding the reliance on an inflectional lexicon. For this purpose, we compare two lemmatization approaches:

- The baseline approach - here the large language model is first fine-tuned for the task of morphosyntactic tagging. MSD predictions obtained from the trained model are then used as input for an inflectional lexicon in order to perform word lemmatization.
- The proposed approach - here the large language model is fine-tuned for the task of predicting string edit tags which encode the transformations necessary to turn a given surface token into its lemma. Lemmatization is performed by directly applying the predicted string edits to each token.

In order to train and evaluate our models, we use four datasets:

1. The Serbian linguistic training corpus SETimes.SR 2.0 - contains around 100,000 tokens of newswire texts (Batanović et al., 2023)

2. The Croatian linguistic training corpus hr500k 2.0 - contains about 500,000 tokens of texts from different genres, including newswire, blog posts, messages from online forums, etc. (Ljubešić & Samardžić, 2023)

3. The Serbian Twitter training corpus ReLDI-NormTagNER-sr 3.0 - contains around 100,000 tokens of Twitter texts (Ljubešić et al., 2023b)

4. The Croatian Twitter training corpus ReLDI-NormTagNER-hr 3.0 - contains around 100,000 tokens of Twitter texts (Ljubešić et al., 2023a)

All four datasets have been manually annotated for a variety of NLP tasks, including morphosyntactic tagging and lemmatization, but none of them have previously been used to evaluate the proposed lemmatization approach. Our training and evaluation process was conducted in two settings: one using the predefined train-dev-test data splits in each of the datasets, and another using 10-fold cross-validation. For both approaches we evaluated model performances after multiple fine-tuning lengths, ranging from 1 to 25 epochs.

For the baseline approach we first use the train data gold MSD tags to fine-tune BERTić on the task of morphosyntactic tagging. We then use its output on the test data and the test data surface tokens to query an inflectional lexicon and obtain lemma predictions. Similarly to (Terčon & Ljubešić, 2023), we use the hrLex 1.3 inflectional lexicon for Croatian, and the srLex 1.3 lexicon for the Serbian data. The hrLex 1.3 lexicon contains 164,206 entries, while srLex 1.3 contains 169,328 entries. The lookup function is implemented to be robust, and it functions as a sieve. It first checks whether the lexicon has an entry which fits the lookup constraints completely, meaning that it has an exact match for both the token and the provided morphosyntactic tag. If this exact lookup does not yield any results, the lookup function checks whether the lexicon contains an entry with different token capitalization variants (lowercase, uppercase and all caps), and the exact match for the MSD tag. If this lookup also does not prove successful, the next step is to search for the entry which has the exact same token, but where only the part-of-speech is matched, rather than the whole morphosyntactic tag. Again, a failed lookup for the exact token is followed by trying different token capitalization variants. Finally, if none of the attempts prove fruitful, the last lookup is conducted only based on the token, disregarding the morphosyntactic tag altogether. If the token does not exist in the lexicon in any shape or form, the lemmatizer will simply assume that the lemma is the same as token (uppercased if the morphosyntactic tagger classifies it as proper noun). Additionally, there are several rules implemented to handle punctuation and abbreviations.

For the proposed new approach, the transformation tuples i.e. string edits are created for each word using the method proposed by (Gesmundo & Samardžić,

2012a). The first step is to extract the longest common substring between the token and the lemma. If the token and lemma have no common substring, the set of necessary string edits can be arbitrarily defined. For example in the case of token 'was' and the lemma 'be', the tuple could be [3, 'be', 0, 0] or [0, 0, 3, 'be']. In this case we have opted for the first possibility, so all the changes are modelled as suffix changes. If the token and its respective lemma have a common substring of at least one character, the procedure is as follows: the part of the token which comes after the longest common substring is considered as the suffix which needs to be removed for lemmatization, while the part of the lemma which comes after the longest common substring is considered as the suffix which needs to be added. The same logic is applied to define the prefix transformations. The pre-trained BERTić model is then fine-tuned on the task of per-token tag prediction, where the tags are defined by the transformation tuples (string edits).

4 DATASET ANALYSIS

It is worth noting that the number of distinct tags varies greatly between the datasets. In SETimes.SR 2.0 there are 310 tags, or different ways in which a token is transformed into its respective lemma. The number of distinct string edits is almost twice as large in hr500k 2.0, with 597 tags, which is a consequence of the Croatian dataset being five times larger than the Serbian one. However, the highest number of different tags is found in ReLDI-NormTagNER-hr 3.0 which has 2056 distinct string edits, with 1314 of them being singleton (appearing for only one token-lemma pair). Similarly, ReLDI-NormTagNER-sr 3.0 contains 1825 distinct tags, with over 1100 singletons. For comparison, SETimes.SR 2.0 has only 72 singleton tags, and hr500k 2.0 has 151. It is evident that the number of distinct string edit tags grows rapidly when working with non-standard textual data. In our experiments we treat all string edit tags equally and leave for future consideration the issue of data sparsity resulting from a large number of singleton tags in non-standard language.

Table 1 presents an overview of the most frequent string edit tags in the four datasets we considered. While there are significant differences in the number of distinct tags between the datasets, particularly singleton ones, there are only slight differences in the ten most frequent tags in each corpus.

Table 1: Most frequent string edit tags.

	SETimes.SR 2.0	ReLDI-NormTagNER-sr 3.0	hr500k 2.0	ReLDI-NormTagNER-hr 3.0
1	[0, ”, 0, ”]	[0, ”, 0, ”]	[0, ”, 0, ”]	[0, ”, 0, ”]
2	[1, ”, 0, ”]	[1, ”, 0, ”]	[1, ”, 0, ”]	[1, ”, 0, ”]
3	[1, ’a’, 0, ”]	[1, ’a’, 0, ”]	[1, ’a’, 0, ”]	[1, ’a’, 0, ”]
4	[2, ’biti’, 0, ”]	[1, ’ti’, 0, ”]	[2, ’biti’, 0, ”]	[2, ’biti’, 0, ”]
5	[2, ’ti’, 0, ”]	[2, ’biti’, 0, ”]	[2, ’ti’, 0, ”]	[1, ’ti’, 0, ”]
6	[1, ’i’, 0, ”]	[0, ’ti’, 0, ”]	[1, ’i’, 0, ”]	[0, ’ti’, 0, ”]
7	[2, ”, 0, ”]	[2, ’ti’, 0, ”]	[2, ”, 0, ”]	[2, ’ti’, 0, ”]
8	[0, ’ti’, 0, ”]	[0, ’be’, 0, ”]	[0, ’ti’, 0, ”]	[1, ’i’, 0, ”]
9	[2, ’an’, 0, ”]	[1, ’i’, 0, ”]	[1, ’ti’, 0, ”]	[0, ’be’, 0, ”]
10	[0, ’be’, 0, ”]	[2, ”, 0, ”]	[0, ’be’, 0, ”]	[2, ”, 0, ”]

By far the most frequent tag is the one which indicates that nothing is to be done to the token in order to lemmatize it, i.e. the token is already in the lemma form. Most of the tags are expected because they add either infinitive suffix (’-ti’) or typical nominal suffixes like ’-a’ for nouns of feminine gender or ’-an’ for adjectives. The tag [2, ’biti’, 0, ”] covers almost all cases of lemmatization for tokens which are conjugations of the verb ’to be’ (’biti’). The only tag which might not be intuitively understood is [0, ’be’, 0, ”] because ’-be’ is not a typical suffix in Serbian or Croatian. However, this tag explains/encodes the lemmatization for token ’se’ (whose lemma is ’sebe’) and is a very frequent reflexive pronoun, making this tag quite prominent in all four datasets.

We can also notice that none of the most frequent string edits have information encoded regarding the prefix. This is not particularly surprising, since both Croatian and Serbian store most of the inflective information in the suffixes. However, a substantial portion of string edit tags do have prefix transformations encoded, with the ratio ranging between 20% in SETimes.SR (60 out of 310 distinct tags) and 40% in both ReLDI-NormTagNER-sr 3.0 and ReLDI-NormTagNER-hr 3.0 datasets. This difference in the frequency of prefix encodings is probably due to the irregularities in the Twitter data and the fact that the string edits produced for these two datasets often (accidentally) include spelling corrections and re-diacritization. A detailed breakdown of the distri-

bution of string edit tags and their characteristics across the four datasets we consider is shown in Table 2.

Table 2: Distribution of string edit tags and their characteristics across the four datasets.

	SETimes.SR 2.0	hr500k 2.0	ReLDI-NormTagNER-sr 3.0	ReLDI-NormTagNER-hr 3.0
Suffix only	249	419	1078	1226
Prefix only	3	11	180	255
Both	57	166	566	574

The predicted sets of necessary string edits are evaluated by verifying whether the token edited in the proposed way really does convert into the lemma. This verification generally confirmed that the transformations are properly produced, but in ReLDI-NormTagNER-sr and ReLDI-NormTagNER-hr there are a number of cases where applying the proposed set of string edits did not correctly create the expected lemma. This occurs in 799 tokens in the ReLDI-NormTagNER-sr dataset, and in 931 cases in the ReLDI-NormTagNER-hr data. The reason why this happens only in the non-standard data is because the original tokens here sometimes have misspellings or are written without diacritic marks, so in these cases the lemmatization process should entail token normalization as a first step. Text written incorrectly and/or without diacritics is quite common in both Serbian and Croatian web corpora, so it is important to have a strategy to deal with this issue when working with text from online sources. While fixing spelling errors is not suitable to be defined using string edits, the issue of undiacriticized text could potentially be addressed with a preprocessing step using a dedicated tool (Ljubešić et al., 2016) in cases where this is an evident problem.

In all four datasets combined there are 3391 different string edit tags. Their overlap of can be seen in Figure 1. In the diagram, we can see that the overlap of string edit tags between all four datasets is 215. Since SETimes.SR has the smallest number of distinct tags (310), we can conclude that the overlap is proportionately quite high. The highest overlap count can be found between the ReLDI-NormTagNER-hr and ReLDI-NormTagNER-sr data. This is a consequence of these two datasets both having a much higher number of different tags than the other two. We can conclude that the non-standard tokens

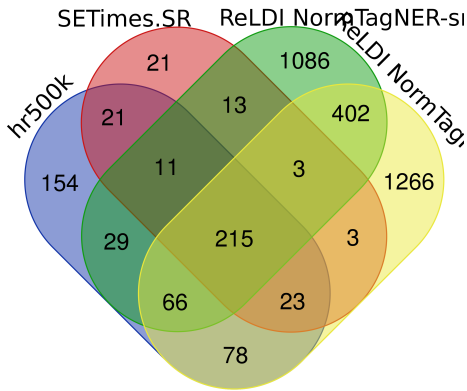

Figure 1: Venn diagram of string edit tags in the four datasets.

which are encountered in tweets increase the number and variety of string edit tags far more than the size of the dataset, since hr500k which is five times larger than all the other datasets has only 154 tags that do not appear in other datasets, while ReLDI-NormTagNER-hr and ReLDI-NormTagNER-sr have 1266 and 1066 respectively.

5 EVALUATION RESULTS AND DISCUSSION

Table 3 contains the evaluation results for all four datasets. The best results in each evaluation setting are shown in boldface.

As mentioned in the previous section, the lookup function used in the baseline model is not trivial. This explains why the model based on morphosyntactic tagging and lexicon lookup performs better across all datasets for a smaller number of epochs. Basically, the errors in morphosyntactic tagging are being compensated by the robust lookup function. However, when models are fine-tuned for ten or more epochs, the approach based on string transformations noticeably outperforms the 'standard' baseline model. On no datasets does the baseline model accuracy increase by more than 1-1.5% through additional training, and these improvements can even be as low as 0.15% on hr500k. On the other hand, our proposed approach typically shows clear accuracy improvements as the number of fine-tuning epochs is increased, and only on hr500k does it reach a performance plateau after 10-15 epochs.

Table 3: Results of model evaluation.

	Epochs	*10-fold CV*		*train-dev-test*		
		MSD + Lexicon	String edits	MSD + Lexicon	String edits	CLASSLA
SETimes.SR	1	95.1	84.96	94.9	81.23	98.02
	5	96.14	95.77	95.98	95.57	
	10	96.2	97.23	96.06	97.03	
	15	96.24	97.65	96.13	97.36	
	20	96.23	97.81	96.06	97.58	
	25	96.24	**97.86**	96.07	**97.76**	
hr500k	1	96.27	94.64	96.28	94.33	98.02
	5	96.41	98.1	96.48	98.16	
	10	96.4	98.38	96.5	98.54	
	15	96.4	98.43	96.5	98.58	
	20	96.4	**98.44**	96.5	**98.63**	
	25	96.41	98.43	96.5	98.62	
Reldi SR	1	85.78	76.86	86.25	76.04	94.92
	5	87.02	90.07	87.46	89.57	
	10	87.1	92.64	87.6	92.30	
	15	87.13	93.63	87.65	93.36	
	20	87.11	94.12	87.61	93.72	
	25	87.11	**94.39**	87.63	**94.06**	
Reldi HR	1	85.44	79.3	85.48	76.88	93.36
	5	86.33	90.31	86.48	89.9	
	10	86.38	92.3	86.57	91.67	
	15	86.41	92.98	86.56	92.58	
	20	86.40	93.43	86.59	93.04	
	25	86.40	**93.64**	86.64	**93.13**	

Both models perform significantly worse when trained and evaluated on Twitter data, for both Croatian and Serbian. This is, of course, due to the fact that these datasets contains non-standard language, so they can be expected to contain many out-of-vocabulary words, as well as unexpected symbols, non-standard punctuation uses and spelling errors. All of these factors have a significantly greater negative effect on the lexicon-based approach, which attains around 10% lower accuracy scores on the non-standard language datasets than on the standard ones. Conversely, the performance of the model based on string edits is only around 5% lower on the non-standard data, which indicates that this approach is more adaptable to datasets which are further from the linguistic norm. Nevertheless, the same trend is noticeable as in the previous two datasets: the approach based on morphosyntactic tagging and lexicon lookup performs better when the model is trained for a small number of epochs, but it is easily outperformed by the approach based on string edit prediction when the fine-tuning length is increased.

In order to see to what extent the 'traditional' model is affected by the lexicon we have conducted an analysis of the predictions done by the models on different datasets, and this can be seen in Table 4. We have classified the errors in three groups: out-of-vocabulary words; tokens which exist in the lexicon but whose lexicon lemmas are different from the gold standard in the datasets; and cases of ambiguity where the token exists in the lexicon with multiple lemmas, one of which is equivalent to the dataset gold standard, but the model makes a mistake by selecting a different lemma/meaning. We can see that in all the cases, the majority of issues could not be avoided with model improvement because they are lexicon related.

Table 4: Error distribution for lemmatization models relying on inflectional lexicons.

	SETimes.SR	hr500k	ReldiSR	ReldiHR
Out-of-vocabulary	26%	33%	72%	59%
Lexicon issues	36%	26.5%	17%	13%
Ambiguity issues	38%	40.5%	11%	28%

Even though the results of lemmatization on all four datasets vary substantially, they all follow the same two main patterns. Firstly, extended fine-tuning of the baseline model never yields more than 1.5% accuracy improvement. This is to

a certain extent a consequence of the robust lookup function, but it also indicates that the lemmatization models based on morphosyntactic tagging and inflectional lexicons have an inherent limitation in performance, likely due to the size and scope limitations of the inflectional lexicons themselves. Secondly, after a certain number of epochs the models based on predicting string edit tags outperform the lexicon-based models, despite the advanced lexicon lookup function. This indicates that defining lemmatization as a string edit prediction task in the proposed way may truly be more suitable for large language models. We also note that the results obtained using 10-fold cross-validation and those based on the provided train, development, and test dataset splits do not vary drastically. The main difference is that the evaluation via predefined splits tends to slightly overestimate the performance of the baseline model and underestimate the performance of the string edit model on most datasets, when compared to CV results.

In order to compare our models with the state-of-the-art, we can look at the results of the CLASSLA-Stanza models on the test portions of the four datasets. For SETimes.SR, the authors report a score of 98.02% while our string edit based model has an accuracy score of 97.76%. On the other hand, when trained and evaluated using hr500k, the CLASSLA-Stanza model scores 98.02% while the model based on string edits outperforms it with a score of 98.63% (Terčon & Ljubešić, 2023). This seems to indicate that the string edit based model benefits more from a larger dataset, although tests on additional such datasets would be required to firmly validate this conclusion, since hr500k was the only larger dataset at our disposal.

When it comes to non-standard data, we can see that although our baseline model performs significantly worse than CLASSLA-Stanza lemmatizer model, the model based on string edits performs comparably well. CLASSLA-Stanza achieves a score of 94.92% on ReLDI-NormTagNER-sr dataset, while the model based on string edits reaches 94.06%. On the ReLDI-NormTagNER-hr dataset, the model based on string edits achieves a score of 93.13%, while CLASSLA-Stanza reaches 93.36%.

Considering the fact that we have not performed hyperparameter optimization, it is expected that CLASSLA-Stanza lemmatizers achieve better scores on most of the datasets. Also, it is important to keep in mind that the lemmatizer mod-

els for non-standard language in the CLASSLA-Stanza package were trained on combined non-standard and standard data, and expanding the size and scope of the training dataset in this way can significantly improve the model performance. As far as model complexity is concerned, while we have not performed measurements of computational and energy requirements of CLASSLA-Stanza vs. our proposed apporach, we estimate that they are roughly similar, since the BERTić model used in our experiments is, by current standards, a relatively small LLM.

6 CONCLUSION

In this paper we have compared two lemmatization approaches for Serbian and Croatian, with the goal of assessing whether tackling lemmatization as a string edit tag prediction task would prove to be better than the 'standard' approach of relying on a morphosyntactic tagging model and an inflectional lexicon. The necessary string edits, which explain how the token can be transformed into its lemma, are encoded in the forms of tuples as proposed by (Gesmundo & Samardžić, 2012a). We have shown that even with a robust lookup function, lemmatization models based on morphosyntactic tagging are being outperformed by the models which learn to lemmatize by tagging tokens based on their necessary string edits. These results are consistent for both the newswire and the Twitter domain, as well as for both Serbian and Croatian data.

In the future we aim to verify these findings on other, specialized domains, such as legal texts, and perform cross-domain and cross-dataset evaluations. We will also examine the impact of the proposed lemmatization approach on different pronunciations of Serbian (Ekavian vs Ijekavian). Another possibility for model improvement would be to combine the datasets and train the models on a larger number of tokens, possibly even cross-linguistically.

7 ACKNOWLEDGMENTS

This work was supported by the COMtext.SR project. [1]

[1] https://github.com/ICEF-NLP/COMtext.SR

REFERENCES

Batanović, V., Cvetanović, M., & Nikolić, B. (2018). Fine-grained semantic textual similarity for Serbian. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).* European Language Resources Association (ELRA). https://aclanthology.org/L18-1219

Batanović, V., Ljubešić, N., Samardžić, T., & Erjavec, T. (2023). *Serbian linguistic training corpus SETimes.SR 2.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1843

Batanović, V., & Nikolić, B. (2017). Sentiment classification of documents in serbian: The effects of morphological normalization and word embeddings. *Telfor Journal*, *9*(2), 104-109. https://doi.org/10.5937/telfor1702104B

Bergmanis, T., & Goldwater, S. (2018). Context sensitive neural lemmatization with lematus. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 1391–1400). Association for Computational Linguistics. http://aclweb.org/anthology/N18-1126

Berkecz, P., Orosz, G., Szántó, Z., Szabó, G., & Farkas, R. (2023). *Hybrid lemmatization in huspacy.* Arxiv. *http://dx.doi.org/10.48550/arXiv.2306.07636*

Chrupala, G. (2006). Simple data-driven context-sensitive lemmatization. *Procesamiento del Lenguaje natural, Revista* (37), 121-127.

Chrupala, G., Dinu, G., & van Genabith, J. (2008). Learning morphology with morfette. In *Proceedings of the international conference on language resources and evaluation, LREC 2008, 26 May - 1 June 2008.* http://www.lrec-conf.org/proceedings/lrec2008/summaries/594.html

Dreyer, M., Smith, J., & Eisner, J. (2008). Latent-variable modeling of string transductions with finite-state methods. In *2008 conference on empirical methods in natural language processing, EMNLP 2008, proceedings of the conference, 25-27 october 2008, honolulu, hawaii, usa, A meeting of sigdat, a special interest group of the ACL* (pp. 1080–1089). http://www.aclweb.org/anthology/D08-1113

Eger, S., Gleim, R., & Mehler, A. (2016). Lemmatization and morphological tagging in german and latin: A comparison and a survey of the state-of-the-art. In *Proceedings of the tenth international conference on language resources and evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.* http://www.lrec-conf.org/proceedings/lrec2016/summaries/656.html

Garabík, R., & Mitana, D. (2022). Accuracy of slovak language lemmatization and msd tagging – morphodita and spacy. *LLOD Approaches for Language Data Research and Management LLODREAM2022 : International Scientific Interdisciplinary Con-*

ference, September 21-22, 2022 : Abstract Book. https://cris.mruni.eu/cris/handle/007/18680

Gesmundo, A., & Samardžić, T. (2012a). Lemmatisation as a tagging task. In *50th annual meeting of the association for computational linguistics, acl 2012 - proceedings of the conference* (pp. 368-372). https://aclanthology.org/P12-2072/

Gesmundo, A., & Samardžić, T. (2012b). Lemmatising serbian as category tag- ging with bidirectional sequence classification. In *Proceedings of the eight international conference on language resources and evaluation (LREC'12).* European Language Resources Association (ELRA).

Jursic, M., Mozetic, I., Erjavec, T., & Lavrac, N. (2010). Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *J. UCS*, *16*(9), 1190–1214. https://doi.org/10.3217/jucs-016-09-1190

Kondratyuk, D., Gavenčiak, T., Straka, M., & Hajič, J. (2018). Lemmatag: Jointly tagging and lemmatizing for morphologically rich languages with brnns. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4921–4928). Association for Computational Linguistics. http://aclweb.org/anthology/D18-1532

Koskenniemi, K. (1984). A general computational model for word-form recognition and production. In *10th international conference on computational linguistics and 22nd annual meeting of the association for computational linguistics, proceedings of COLING '84, July 2-6, 1984, Stanford University, California, USA.* (pp. 178–181). http://aclweb.org/anthology/P84-1038

Kutuzov, A., & Kuzmenko, E. (2019). To lemmatize or not to lemmatize: how word normalisation affects elmo performance in word sense disambiguation. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing. (pp. 22–28). https://aclanthology.org/W19-6203/*

Ljubešić, N. (2019a). *Inflectional lexicon hrLex 1.3.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1232

Ljubešić, N. (2019b). *Inflectional lexicon srLex 1.3.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1233

Ljubešić, N., Erjavec, T., Batanović, V., Miličević, M., & Samardžić, T. (2023a). *Croatian twitter training corpus ReLDI-NormTagNER-hr 3.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/ 1793

Ljubešić, N., Erjavec, T., Batanović, V., Miličević, M., & Samardžić, T. (2023b). *Serbian twitter training corpus ReLDI-NormTagNER-sr 3.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/ 1794

Ljubešić, N., Erjavec, T., & Fišer, D. (2016). Corpus-based diacritic restoration for South Slavic languages. In N. Calzolari et al. (Eds.), *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)* (pp. 3612–3616). European Language Resources Association (ELRA).

https://aclanthology.org/L16-1573

Ljubešić, N., & Lauc, D. (2021). BERTić - the transformer language model for Bosnian, Croatian, Montenegrin and Serbian. In *Proceedings of the 8th work- shop on balto-slavic natural language processing* (pp. 37–42). Association for Computational Linguistics. https://www.aclweb.org/anthology/ 2021.bsnlp-1.5

Ljubešić, N., & Samardžić, T. (2023). *Croatian linguistic training corpus hr500k 2.0.* Slovenian language resource repository CLARIN.SI. http://hdl.handle.net/11356/1792

Nicolai, G., & Kondrak, G. (2016). Leveraging inflection tables for stemming and lemmatization. In *Proceedings of the 54th annual meeting of the association for computational linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long papers.* http://aclweb.org/anthology/P/P16/P16-1108.pdf

Plisson, J., Lavrac, N., & Mladenic, D. (2004). A rule based approach to word lemmatization. In *Proceedings of is04. https://aile3.ijs.si/dunja/SiKDD2004/Papers/Pillson-Lematization.pdf*

Terčon, L., & Ljubešić, N. (2023). *Classla-stanza: The next step for linguistic processing of south slavic languages.* Arxiv. *https://doi.org/10.48550/arXiv.2308.04255*

LEMATIZACIJA SRBSKEGA IN HRVAŠKEGA JEZIKA Z UPORABO STRING EDIT PREDICTION

V tem prispevku preučujemo učinkovitost lematizacije besedil v srbščini in hrvaščini z uporabo vnaprej usposobljenega velikega jezikovnega modela, natančno nastavljenega na nalogo predvidevanja urejanja niza. Lematizacijo definiramo kot nalogo označevanja, kjer je vsaka transformacija besede-leme predstavljena kot oznaka za urejanje niza, ki kodira potrebne spremembe predpone in pripone. Naš pristop je preverjen z uporabo velikega jezikovnega modela BERTić in vodi do izboljšanih rezultatov na standardnem srbskem SETimes.SR in standardnem hr500k hrvaškem naboru podatkov, ter na naborih podatkov ReLDI-NormTagNER-sr in ReLDI-NormTagNER-hr. Njegova dodatna prednost je, da se ne zanaša na nobene leksikalne baze podatkov, zaradi česar je enostavno uporaben za različna besedilna področja in jezikovne različice.

Keywords: Lemmatization, BERTić, SETimes.SR, hr500k, ReLDI-NormTagNER-sr, ReLDI-NormTagNER-hr