Sentiment Classification of Documents in Serbian: The Effects of Morphological Normalization and Word Embeddings

Vuk Batanović and Boško Nikolić

***Abstract* — An open issue in the sentiment classification of texts written in Serbian is the effect of different forms of morphological normalization and the usefulness of leveraging large amounts of unlabeled texts. In this paper, we assess the impact of lemmatizers and stemmers for Serbian on classifiers trained and evaluated on the Serbian Movie Review Dataset. We also consider the effectiveness of using word embeddings, generated from a large unlabeled corpus, as classification features.**

***Keywords* — comparative evaluation, lemmatization, morphology, sentiment analysis, stemming, word embeddings.**

I. Introduction

SENTIMENT analysis is the problem of automatically assessing the sentiment of a given text. In sentiment classification, the basic task in sentiment analysis, the aim is to classify the text as positive or negative, with the occasional inclusion of the neutral class. Long documents are easier to classify than short texts, as their overall meaning depends less on syntactic specificities and figures of speech. The classification problem is usually solved using machine-learning algorithms, mainly supervised ones. However, a sufficiently large set of texts, annotated according to their sentiment, is required in order to train and evaluate classifiers.

The first sentiment analysis systems for English based on machine learning were created about 15 years ago [1]. The current state-of-the-art is mostly focused on deep-learning models trained on very large datasets [2]. Sentiment classification in other languages, particularly minor ones, has been slower to develop, due to the difficulty in procuring the necessary text corpora. The first and, thus far, the only publicly available sentiment analysis dataset in Serbian is the Serbian Movie Review Dataset – *SerbMR*[1] [3]. It comes in two variants – SerbMR-2C (ISLRN 016-049-192-514-1), containing only the positive and the negative examples, and SerbMR-3C (ISLRN 229-533-271-984-0), which also includes the neutral ones. Each sentiment class in SerbMR contains 841 reviews.

This paper explores the extent to which the existing morphological normalization tools for Serbian can affect the performance of document sentiment classifiers. We also assess the use of additional large unlabeled corpora via simple word embedding-based techniques, and the value of morphological normalization in this context. Recently, Rotim and Šnajder [4] presented a similar study for short-text sentiment classification in Croatian, but they only evaluated a single stemmer and lemmatizer and did not consider the effect of stemming on word embeddings.

The remainder of the paper is structured as follows: we first give an overview of morphological normalization methods and a survey of the available morphological tools for Serbian. We also describe the method and the resources used to create word embeddings for Serbian. We then evaluate and discuss the effects of morphological normalization tools on the bag-of-words/n-grams models for sentiment classification of documents in Serbian, in the binary and the multiclass setting. A similar evaluation is performed for models that use word embeddings. Lastly, we consider some points worthy of further research.

II. Morphological Normalization

Morphological normalization is the merger of different morphological variations of a term into the same base form. The role of morphological normalizers in the sentiment analysis of morphologically rich but resource-deficient languages like Serbian is to lower the vocabulary size and thereby reduce data sparsity, which makes it easier for classifiers to accurately model the impact of each word or expression. Stemming and lemmatization are two commonly used normalization procedures.

Stemming removes the suffixes of a word, resulting in its *stem*, and does not generally distinguish between inflectional and derivational morphological changes.

Paper received September 3, 2017; revised November 27, 2017; accepted December 1, 2017. Date of publication December 25, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Miroslav Lutovac.

This paper is a revised and expanded version of the paper presented at the 24th Telecommunications Forum TELFOR 2016 [34].

This work was partially supported by the III44009 research project of the Ministry of Education, Science and Technological Development of the Republic of Serbia.

Vuk Batanović is with the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia (e-mail: vuk.batanovic@student.etf.bg.ac.rs).

Boško Nikolić is with the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia (e-mail: nbosko@etf.bg.ac.rs).

[1] http://vukbatanovic.github.io/SerbMR/

Stemmers can sometimes understem or overstem, removing too little or too much of the word ending. This can result in errors where words with completely different semantics are conflated into one stem, e.g. when reducing the words *general*, *generation*, and *generator* to *gener*.

Lemmatization aims to replace the given word with its *lemma*, or dictionary form, which limits its effect to inflectional morphology and prevents the occurrence of errors typical of stemmers. However, unlike stemming, which does not require any information aside from the word to be stemmed, lemmatization relies on word context – lemmatizers usually presuppose that the text is already marked with part-of-speech (POS) tags. Both tagging and lemmatization are often tackled as a sequence prediction problem. Hence, obtaining the final lemmatized text can be a much more time-consuming process than stemming, which is usually implemented as a simple list of automatically or manually compiled transformation rules.

III. Stemmers and Lemmatizers for Serbian

A. Stemmers

We have found three publicly available stemming algorithms for Serbian and one for Croatian, which is also applicable to Serbian. The optimal and the greedy stemmer of Kešelj and Šipka [5], and the improved version of the greedy algorithm by Milošević [6] all employ a suffix-subsumption approach, while the stemmer for Croatian by Ljubešić and Pandžić[2], which is a refinement of the algorithm presented in [7], relies on regular expressions.

Batanović et al. [3] reimplemented all these algorithms as a unified stemming package – *SCStemmers*[3] – and evaluated their usefulness in sentiment classification. Despite being somewhat slower than the other algorithms, due to its use of regular expressions, the stemmer of Ljubešić and Pandžić provided the greatest increase in classifier performances on this task.

B. Lemmatizers

In this paper, we have considered two publicly available lemmatizers for Serbian and one for Croatian. All of them are accompanied by a POS tagger module.

Gesmundo and Samardžić presented two versions of *BTagger*[4], a system that performs lemmatization as a category tagging task – one where only the word suffixes are normalized [8], and one which also deals with word prefixes, allowing for full lemmatization [9]. Agić et al. developed a lemmatization model for Croatian[5] which was also successfully applied to Serbian [10]. They evaluated several lemmatization tools and concluded that the *CST* lemmatizer [11] achieves the highest accuracy with their model. Continuing this line of work, Ljubešić et al. presented a lemmatizer for Serbian[6] that relies on a large inflectional lexicon and an improved POS tagger [12].

Aside from the three lemmatizers evaluated here, there are a few other publicly available packages that could be used for lemmatizing texts written in Serbian. However, they were discarded from evaluation since previous work showed them to be inferior to the aforementioned algorithms. Gesmundo and Samardžić found that *LemmaGen* of Juršič et al. [13] performs significantly worse when lemmatizing Serbian than their own approach [8]. Similarly, Agić et al. found the chosen CST lemmatizer to be better than the *PurePos* [14] and the *TreeTagger* [15] libraries, when used with their model.

IV. Word Embeddings

A word embedding is a dense, low-dimensional, real-valued vector whose dimensions represent latent features of a word [16]. These features capture the syntactic and the semantic properties of a word, making embeddings a simple yet powerful method of word representation in downstream natural language processing tasks.

Word embeddings are generated from a large text corpus, most often an unlabeled one. To create the Serbian word embeddings we used the largest freely available collection of texts in Serbian we could find – the Serbian web corpus (*srWaC*)[7], which contains 555 million tokens collected from the *.rs* top-level domain [17].

In this paper we have utilized *Word2Vec* [18], [19], one of the most popular word embedding algorithms, as implemented in the *Gensim* library [20]. We have focused on the skip-gram *Word2Vec* model, in which embeddings are produced by a shallow neural network trained to reconstruct the context of the given words. The alternative CBOW model, in which a network is trained to predict a word given its context, is not included in our evaluation since preliminary experiments found the skip-gram embeddings superior on the sentiment classification task.

V. Evaluation

The first part of the evaluation is performed on SerbMR-2C and SerbMR-3C using the WEKA (*Waikato Environment for Knowledge Analysis*) workbench [21] and a bag-of-words/n-grams approach, in which a document is modeled as an unordered set of words/n-grams. We consider two basic classifiers popular in the sentiment analysis literature – Multinomial Naïve Bayes (MNB) and Support Vector Machines (SVM). On the binary task we also evaluate NBSVM, a mixture of these two algorithms that was shown to work well in binary settings [3], [22]. The implementations we utilize are WEKA's default version of MNB, LIBLINEAR's SVM [23], and Batanović et al.'s implementation of NBSVM for WEKA[8] [3]. As suggested by Wang and Manning [22] we employ the L2 regularization and loss function for SVM and NBSVM. To ensure high test result replicability [24], [25] we evaluate using the same 10-run-average of 10-fold cross-validation as in [3]. The SVM and NBSVM hyperparameters are also optimized as in [3], through nested cross-validation.

[2] http://nlp.ffzg.hr/resources/tools/stemmer-for-croatian/
[3] http://vukbatanovic.github.io/SCStemmers/
[4] http://clcl.unige.ch/btag/
[5] http://nlp.ffzg.hr/resources/models/tagging/
[6] http://reldi.spur.uzh.ch/blog/croatian-and-serbian-lemmatiser/
[7] http://reldi.spur.uzh.ch/blog/serbian-web-corpus/
[8] http://vukbatanovic.github.io/NBSVM-Weka/

TABLE 1: BAG-OF-WORDS MODEL – CLASSIFIER CV ACCURACIES ON SERBMR-2C: POSITIVE/NEGATIVE

Morphological normalization	MNB	SVM	NBSVM
No normalization	80.18	82.00	83.50
Stemmers			
Kešelj & Šipka – optimal	**81.32**	83.32	84.01
Kešelj & Šipka – greedy	80.45	83.16	83.73
Milošević	81.04	**83.49**	**84.74**
Ljubešić & Pandžić	81.23	83.34	84.19
Lemmatizers			
BTagger – suffix	80.78	83.45	82.88
BTagger – suffix + prefix	80.91	83.52	83.04
Agić et al. – CST	80.64	82.69	82.86
Ljubešić et al.	**81.19**	**83.82**	**84.20**

TABLE 2: BAG-OF-WORDS MODEL – CLASSIFIER CV ACCURACIES ON SERBMR-3C: POSITIVE/NEUTRAL/NEGATIVE

Morphological normalization	MNB	SVM
No normalization	58.22	60.86
Stemmers		
Kešelj & Šipka – optimal	58.65	61.68
Kešelj & Šipka – greedy	58.04	60.92
Milošević	58.20	61.86
Ljubešić & Pandžić	**58.96**	**62.05**
Lemmatizers		
BTagger – suffix	57.97	61.61
BTagger – suffix + prefix	**58.16**	61.45
Agić et al. – CST	57.65	61.07
Ljubešić et al.	57.62	**61.97**

In order to focus on the issue of morphological normalization we adopt the optimal settings for negation marking and for the choice of machine learning features and their types, in both the binary and the multiclass task, from [3], with two exceptions. Firstly, instead of the default WEKA tokenizer used in the previous classifier evaluations, we employ the tokenizer for Serbian included in the ReLDI (*Regional Linguistic Data Initiative*) project repository[9] [26], [27] and we retain only the alphanumerical tokens as input to the classifiers. In addition, we use binary features for MNB and NBSVM, since it was shown they are more suited to these classifiers [3], [22]. For the SVM we keep the token count features as they work better with classical discriminative algorithms [3]. We view the results obtained by utilizing these settings, without applying any morphological normalization, as a baseline. A paired corrected resampled *t*-test [25] is used to statistically compare the results of the morphologically normalized models with the baseline.

The second part of the evaluation is centered on models that average out the embeddings of the words in a given document and use the resulting mean vector as input to a classifier. We also consider combining the mean-vector input features with the n-gram ones, as well as the impact of morphological normalization on word embeddings. The same ReLDI tokenizer for Serbian and the restriction to alphanumeric tokens are used here as well.

Turian et al. [16] found that the optimal dimensionality of word embeddings depends on the task in question, so we evaluate a number of settings, from 100 dimensions to 1000, and a context window size of either 5 or 10. All other *Word2Vec* parameters are kept at the default *Gensim* settings for the skip-gram architecture. The *srWaC* corpus is parsed to remove punctuation signs, as well as those parts of the corpus that are not in Serbian. We then apply the negation-marking technique from [3] to the remaining text, and we label a single word after each negation word.

Evaluation is performed using the *Scikit-learn* package [28] and the L2-regularized L2-loss LIBLINEAR SVM [23]. Working with dense vectors is computationally expensive, so we limit word-embedding experiments to a single 10-fold cross-validation. A nested 5-fold cross-validation is used to optimize the SVM cost parameter $C \in [10^{-2} - 10^{2}]$. The stochasticity of the skip-gram neural network training produces a small variance in the calculation of word embeddings, making the results in this part of the evaluation slightly approximate.

A. Bag-of-words model

We first evaluate the lemmatizers on a bag-of-words/unigram model, in which the individual words/tokens are used as classifier features. In order to make a fair comparison between the different morphological normalization procedures and sidestep the effects of slightly different preprocessing options utilized here and in [3], we also reevaluate the stemmers for Serbian. The figures for the binary and the multiclass classification are given in Tables 1 and 2.

The results show that using stemming usually leads to classifiers outperforming the baseline, while lemmatization has less consistent effects. Still, no morphologically normalized unigram model demonstrates a statistically significant improvement over the baseline. The overall best stemming algorithms are the one created by Ljubešić and Pandžić (as previously established in [3]), and the one presented by Milošević, whose impact seems boosted by the more precise tokenization algorithm used here. The lemmatizer of Ljubešić et al. is most often the optimal one in terms of its effect on classifier accuracies, yet it still generally fails to surpass the best stemmers.

Such an outcome is probably due to the nature of the two normalization techniques. Stemmers tend to treat inflectional and derivational suffixes in the same manner and thereby conflate not only the inflections of a word, but also many of its derivations. This behavior might not be desirable in some situations, but when training sentiment classifiers with limited resources it actually proves useful, as it allows the model to merge derivationally related words into a single item, thus reducing vocabulary size and data sparsity. Since derivationally related words most often do not express differing sentiments, few classification errors are incurred due to this effect. On the other hand, lemmatizers focus on inflectional morphology only, which limits their vocabulary reduction capability.

[9] http://reldi.spur.uzh.ch/blog/tokeniser/

TABLE 3: VOCABULARY SIZE AS A FUNCTION OF MORPHOLOGICAL NORMALIZATION

Morphological normalization	SerbMR-2C	SerbMR-3C
No normalization	88K	109K
Stemmers		
Kešelj & Šipka – optimal	42K	51K
Kešelj & Šipka – greedy	45K	54K
Milošević	46K	56K
Ljubešić & Pandžić	45K	54K
Lemmatizers		
BTagger – suffix	57K	70K
BTagger – suffix + prefix	56K	69K
Agić et al. – CST	63K	78K
Ljubešić et al.	46K	56K

Table 3 confirms this intuition by showing the vocabulary sizes of the SerbMR dataset[10] after applying different morphological normalization methods. The lemmatizer of Ljubešić et al. is the only one that matches the reduction commonly achieved by stemmers, which partly explains its superiority over the other lemmatization algorithms. Still, even though all the stemmers achieve roughly the same level of vocabulary reduction, they lead to noticeably different classification accuracies. Therefore, it is evident that the inherent quality of the normalization procedure plays an important role as well. In light of these findings, we also experimented with combining the two normalization techniques by applying stemming after lemmatization, but we failed to achieve any consistent improvement in classifier accuracies over a single normalization procedure.

An additional drawback of lemmatizers for Serbian, in the context of sentiment analysis, is that they reduce all degrees of comparison of adverbs and adjectives into one. This behavior can lead to classification errors in the context of a negation – a negation of a positive form is clearly negative (e.g. *not good*), but a negation of a superlative form does not necessarily carry a negative connotation (e.g. *not the best*). Stemmers are unable to generate mistakes of this kind, since the superlative form of adverbs and adjectives in Serbian is generated through the addition of a prefix – "*naj-*".

B. Bag-of-n-grams model

Next, we explore how the addition of bigram and trigram features, i.e. the switch from a bag-of-words to a bag-of-n-grams model, with the rest of the settings fixed, affects classification accuracies. Our aim is not only to compare the impact of different normalization methods, but also to measure the performance limits of bag-of-n-grams models. Hence, we experiment with the strongest algorithms – NBSVM in the binary task and SVM in the multiclass one. We focus on the best normalization tools in each category – the stemmers of Milošević and Ljubešić and Pandžić, and the lemmatizer of Ljubešić et al. in the binary classification, and the same lemmatizer and the latter stemmer in the multiclass task.

Tables 4 and 5 contain the binary and the multiclass classification accuracies. (S) stands for stemmers and (L) for lemmatizers, while *U*, *B*, and *T* denote unigram, bigram, and trigram features, respectively. The differences between the results of normalized and baseline models that are found statistically significant at the 0.05 / 0.01 level are marked with * / **.

In the binary setting, all of the selected normalization tools perform similarly, but the stemmer of Ljubešić and Pandžić manages to be slightly better than the alternatives and raises the maximal recorded accuracy on SerbMR-2C to 86.1% with the *U* + *B* model, due to a better tokenization procedure. In the three-class setting we find that stemming allows for better results than lemmatization, and we observe results similar to those obtained in [3].

TABLE 4: BAG-OF-N-GRAMS MODEL – CLASSIFIER CV ACCURACIES ON SERBMR-2C: POSITIVE/NEGATIVE

Morphological normalization	NBSVM		
	U	*U* + *B*	*U* + *B* + *T*
No normalization	83.50	83.90	83.82
(S) Milošević	84.74	85.97*	85.93*
(S) Ljubešić & Pandžić	84.19	**86.11****	86.01**
(L) Ljubešić et al.	84.20	85.88*	85.84*

TABLE 5: BAG-OF-N-GRAMS MODEL – CLASSIFIER CV ACCURACIES ON SERBMR-3C: POSITIVE/NEUTRAL/NEGATIVE

Morphological normalization	SVM		
	U	*U* + *B*	*U* + *B* + *T*
No normalization	60.86	60.88	60.65
(S) Ljubešić & Pandžić	62.05	**63.02***	62.42
(L) Ljubešić et al.	61.97	62.02	61.50

C. Averaged word embeddings model

Our first goal when working with word embeddings is to determine the optimal vector dimensionality and context window size. We therefore consider a simple model that finds the mean vector for the words present in a document (out-of-vocabulary words are not taken into account), and uses that vector as input to an SVM classifier. We have found that applying TF/IDF and/or NBSVM weighting to word vectors has a detrimental effect, so we do not use it. The results in Table 6 show that larger dimensionalities and window sizes lead to better classification accuracies. However, models that rely solely on averaged word embeddings are easily outperformed by unigram models.

It is possible that a further increase in word embedding dimensionality could yield additional improvements in classification accuracy, but such increases quickly become prohibitively expensive in terms of computational efficiency. On the other hand, our preliminary experiments have shown that increasing the window size beyond the value of 10 does not lead to any consistent gains with regard to classifier performance.

In light of this, we combine the n-gram features with the ones gained through averaging word embeddings, and we measure the effects of the best morphological

[10] The vocabulary size of the non-normalized dataset differs between this paper and [3] due to the use of different tokenization procedures.

TABLE 6: AVERAGED WORD EMBEDDINGS MODEL – SVM CV ACCURACIES

Embedding dimensionality	Window size	SerbMR-2C	SerbMR-3C
100	5	75.57	55.26
100	10	77.29	56.56
300	5	79.13	57.27
300	10	79.37	57.51
500	5	80.74	57.75
500	10	80.80	58.06
1000	5	81.04	58.94
1000	10	**81.92**	**59.25**

normalization tools in each category – the same as in the previous section. We focus on 1000-dimensional embeddings generated with a window size of 10, since they were the optimal ones in the plain embedding-based model, and we use the *Scikit-learn* TF-IDF vectorizer to create the n-gram features. We have found that NBSVM weighting of n-gram features is not beneficial when other, embedding-based features are also given to the classifier. Therefore, both the binary classification results, shown in Table 7, and the multiclass ones, shown in Table 8, are obtained from a standard SVM. *E* in the tables denotes the use of averaged word embedding features, while *U* and *B* stand for unigram and bigram ones. We do not extend the feature set to trigrams since the bag-of-n-grams model evaluation showed that they are not useful for this task.

Our results confirm that on a binary sentiment classification task it is hard to beat the *U* + *B* NBSVM bag-of-n-grams model with simple embeddings-based methods [22]. We find this to be true despite the positive effects of using the selected morphological normalization tools. On the other hand, combining the embedding-based features with the n-gram ones proves to be advantageous in the multiclass setting. Even the baseline *E* + *U* + *B* model outperforms the best bag-of-n-grams approach, while the variant where the Ljubešić and Pandžić stemmer is utilized achieves an accuracy of 64.37%, the maximum recorded on SerbMR-3C. As was the case with the bag-of-n-grams models, stemming again proves superior to lemmatization on the three-class task.

TABLE 7: AVERAGED WORD EMBEDDINGS + BAG-OF-N-GRAMS MODEL – CLASSIFIER CV ACCURACIES ON SERBMR-2C: POSITIVE/NEGATIVE

Morphological normalization	SVM		
	E	*E* + *U*	*E* + *U* + *B*
No normalization	81.92	84.30	85.31
(S) Milošević	81.92	84.36	85.61
(S) Ljubešić & Pandžić	82.93	84.78	85.61
(L) Ljubešić et al.	82.64	85.02	85.73

TABLE 8: AVERAGED WORD EMBEDDINGS + BAG-OF-N-GRAMS MODEL – CLASSIFIER CV ACCURACIES ON SERBMR-3C: POSITIVE/NEUTRAL/NEGATIVE

Morphological normalization	SVM		
	E	*E* + *U*	*E* + *U* + *B*
No normalization	59.25	62.11	63.73
(S) Ljubešić & Pandžić	59.89	62.19	**64.37**
(L) Ljubešić et al.	59.33	61.56	63.62

We have also experimented with other models and methods of using word embeddings on the task of binary sentiment classification, including *Paragraph Vector* [29], *fastText* [30], [31], and a distributed/dense counterpart of NBSVM [32]. However, all of them have failed to outperform the sparse *U* + *B* NBSVM model. These results suggest that the limited amount of data in SerbMR-2C is the key bottleneck, since the aforementioned algorithms have successfully outmatched sparse/bag-of-n-grams methods on the task of sentiment classification of texts in English, where large quantities of training data are available.

D. Normalization efficiency

Another important side of using morphological normalization tools is their efficiency. It is hard to present a comprehensive empirical evaluation of this aspect of the tools since performance figures may vary greatly depending on the available hardware resources and the data in question. Therefore, we make a simple comparison on the task of normalizing the SerbMR-3C dataset on a dual-core 2.0 GHz computer with 8 GB of RAM.

Table 9 contains the approximate execution times, as the exact figures slightly differ from one run to another. Most lemmatizers are several orders of magnitude slower than stemmers, and their efficiency is further reduced due to POS tagging. Only the CST lemmatizer, used by Agić et al., is comparable to stemmers with regard to speed, and it relies on the similarly fast *HunPos* tagger [33].

TABLE 9: EXECUTION TIMES OF MORPHOLOGICAL NORMALIZERS

Morphological normalization	Approximate execution time on SerbMR-3C	
Stemmers		
Kešelj & Šipka – optimal	~5s	
Kešelj & Šipka – greedy	~5s	
Milošević	~7s	
Ljubešić & Pandžić	~35s	
Lemmatizers		
	Lemmatization	POS tagging
BTagger – suffix	~4h 55min	~23min
BTagger – suffix + prefix	~4h 23min	~23min
Agić et al. – CST	~ 7s	~30s
Ljubešić et al.	~2h 6min (for both)	

VI. CONCLUSION

In this paper, we have presented various morphological tools for Serbian and have evaluated their usefulness on the task of document sentiment classification. We have found stemming to be a better option than lemmatization for performing this task in resource-constrained settings, both in terms of classification accuracy and in terms of normalization efficiency. In particular, the stemmer of Ljubešić and Pandžić has proved to be the best contender when utilizing higher order n-gram features.

We have also considered using word embeddings,

generated from a large unlabeled corpus, as classification features. Adding such features to a bag-of-n-grams model can increase classification accuracy in the multiclass setting, but yields no positive effects on the binary task.

Our findings should make it easier to improve classifier results when creating other domain-specific sentiment classification systems for Serbian using limited resources. They may also prove useful for general text classification under similar conditions.

In the future, we plan to verify our results on short-text sentiment classification. We will also extend the evaluation to other semantic tasks, such as semantic similarity.

REFERENCES

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 2002, pp. 79–86.

[2] D. Tang, B. Qin, and T. Liu, "Deep learning for sentiment analysis: successful approaches and future challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 6, pp. 292–303, 2015.

[3] V. Batanović, B. Nikolić, and M. Milosavljević, "Reliable Baselines for Sentiment Analysis in Resource-Limited Languages: The Serbian Movie Review Dataset," in *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, 2016, pp. 2688–2696.

[4] L. Rotim and J. Šnajder, "Comparison of Short-Text Sentiment Analysis Methods for Croatian," in *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing (BSNLP 2017)*, 2017, pp. 69–75.

[5] V. Kešelj and D. Šipka, "A Suffix Subsumption-Based Approach to Building Stemmers and Lemmatizers for Highly Inflectional Languages with Sparse Resources," *INFOtheca*, vol. 9, no. 1–2, p. 23a–33a, 2008.

[6] N. Milošević, "Stemmer for Serbian language." arXiv 1209.4471, 2012.

[7] N. Ljubešić, D. Boras, and O. Kubelka, "Retrieving Information in Croatian: Building a Simple and Efficient Rule-Based Stemmer," in *INFuture2007: Digital Information and Heritage*, Zagreb, Croatia: Department for Information Sciences, Faculty of Humanities and Social Sciences, 2007, pp. 313–320.

[8] A. Gesmundo and T. Samardžić, "Lemmatising Serbian as Category Tagging with Bidirectional Sequence Classification," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, 2012, pp. 2103–2106.

[9] A. Gesmundo and T. Samardžić, "Lemmatisation as a tagging task," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012, pp. 368–372.

[10] Ž. Agić, N. Ljubešić, and D. Merkler, "Lemmatization and Morphosyntactic Tagging of Croatian and Serbian," in *Proceedings of the Fourth Biennial International Workshop on Balto-Slavic Natural Language Processing*, 2013, pp. 48–57.

[11] B. Jongejan and H. Dalianis, "Automatic training of lemmatization rules that handle morphological changes in pre- , in- and suffixes alike," in *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the Fourth International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL 2009)*, 2009, pp. 145–153.

[12] N. Ljubešić, F. Klubička, Ž. Agić, and I.-P. Jazbec, "New Inflectional Lexicons and Training Corpora for Improved Morphosyntactic Annotation of Croatian and Serbian," in *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, 2016, pp. 4264–4270.

[13] M. Juršič, I. Mozetič, T. Erjavec, and N. Lavrač, "Lemmagen: Multilingual Lemmatisation with Induced Ripple-Down Rules," *Journal of Universal Computer Science*, vol. 16, no. 9, pp. 1190–1214, 2010.

[14] G. Orosz and A. Novák, "PurePos 2.0: a hybrid tool for morphological disambiguation.," in *Proceedings of Recent Advances in Natural Language Processing*, 2013, pp. 539–545.

[15] H. Schmid, "Improvements in Part-of-Speech Tagging with an Application to German," in *Proceedings of the ACL SIGDAT-Workshop*, 1995.

[16] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semi-supervised learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, 2010, pp. 384–394.

[17] N. Ljubešić and F. Klubička, "{bs,hr,sr}WaC –Web corpora of Bosnian, Croatian and Serbian," in *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, 2014, pp. 29–35.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Proceedings of the International Conference on Learning Representations Workshop (ICLR 2013)*, 2013.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013)*, 2013, pp. 3111–3119.

[20] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010, pp. 45–50.

[21] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[22] S. Wang and C. D. Manning, "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, 2012, pp. 90–94.

[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[24] R. R. Bouckaert, "Choosing between two learning algorithms based on calibrated tests," in *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*, 2003, pp. 51–58.

[25] R. R. Bouckaert and E. Frank, "Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms," in *Proceedings of the Eighth Pacific-Asia Conference (PAKDD 2004)*, 2004, pp. 3–12.

[26] T. Samardžić, N. Ljubešić, and M. Miličević, "Regional Linguistic Data Initiative (ReLDI)," in *Proceedings of the Fifth Workshop on Balto-Slavic Natural Language Processing (BSNLP 2015)*, 2015, pp. 40–42.

[27] N. Ljubešić, T. Erjavec, D. Fišer, T. Samardžić, M. Miličević, F. Klubička, and F. Petkovski, "Easily Accessible Language Technologies for Slovene , Croatian and Serbian," in *Proceedings of the Conference on Language Technologies & Digital Humanities*, 2016, pp. 120–124.

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

[29] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, 2014, pp. 1188–1196.

[30] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers (EACL 2017)*, 2017, pp. 427–431.

[31] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[32] B. Li, Z. Zhao, T. Liu, P. Wang, and X. Du, "Weighted Neural Bag-of-n-grams Model: New Baselines for Text Classification," in *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, 2016, pp. 1591–1600.

[33] P. Halácsy, A. Kornai, and C. Oravecz, "HunPos – an open source trigram tagger," in *Proceedings of the ACL 2007 Demo and Poster Sessions*, 2007, pp. 209–212.

[34] V. Batanović and B. Nikolić, "Sentiment Classification of Documents in Serbian: The Effects of Morphological Normalization," in *Proceedings of the 24th Telecommunications Forum (TELFOR 2016)*, 2016.