Sentiment Classification of Documents in Serbian: The Effects of Morphological Normalization

Vuk Batanović, Boško Nikolić

Abstract — **Sentiment classification of texts written in Serbian is still an under-researched topic. One of the open issues is how the different forms of morphological normalization affect the performances of different sentiment classifiers and which normalization procedure is optimal for this task. In this paper we assess and compare the impact of lemmatizers and stemmers for Serbian on classifiers trained and evaluated on the Serbian Movie Review Dataset.**

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

I. Introduction

SENTIMENT analysis is the problem of automatically determining and measuring 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. Classifying longer documents is easier than short-text classification, as the overall semantics of longer texts depends less on syntactic specificities and figures of speech. The classification problem is usually solved using machine learning algorithms, predominantly supervised ones. However, in order to train and evaluate classifiers a sufficiently large set of textual examples, annotated according to their sentiment, is required.

The first sentiment analysis systems for English based on machine learning were created about 15 years ago [1]. Development for other languages, particularly minor ones, has been slower, 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] [2]. 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 the SerbMR dataset contains 841 reviews.

In this paper we aim to explore the extent to which the existing morphological normalization tools for Serbian can impact the performance of sentiment classifiers. We also believe this to be the first paper that presents an extrinsic comparative evaluation of almost all of the publicly available morphological tools for Serbian.

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 then evaluate and discuss the effects these tools have on the sentiment classification of documents in Serbian, in the binary and the multiclass setting. Lastly, we consider some points worthy of further research.

II. Morphological Normalization

Morphological normalization deals with merging 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*. Stemming generally does not distinguish between inflectional and derivational morphological changes. Furthermore, stemmers sometimes tend to 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 for proper operation. Performing lemmatization usually presupposes that the text is already marked with part-of-speech (POS) tags, leading to POS taggers and lemmatizers frequently being packaged together. 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 implemented as a simple list of automatically or manually compiled transformation rules.

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/

978-1-5090-4086-5/16/$31.00 ©2016 IEEE

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 [3], and the improved version of the greedy algorithm by Milošević [4] 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 [5], relies on regular expressions.

Batanović et al. [2] 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 [6], and one which also deals with word prefixes, allowing for full lemmatization [7]. Agić et al. developed a lemmatization model for Croatian[5] which was also successfully applied to Serbian [8]. They evaluated the performance of several publicly available lemmatization tools and concluded that the *CST* lemmatizer [9] 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 [10].

It should be noted that, 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. [11] performs significantly worse when lemmatizing Serbian than their own approach [6]. Similarly, Agić et al. found the chosen CST lemmatizer to be better than the *PurePos* [12] and the *TreeTagger* [13] libraries, when used with their model.

IV. Evaluation

Evaluation is performed on SerbMR-2C and SerbMR-3C using the WEKA (*Waikato Environment for Knowledge Analysis*) workbench [14] 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 classification task we also evaluate NBSVM, a combination of these two algorithms that was shown to work well in binary settings [2], [15]. The implementations we utilize are WEKA's default version of MNB, LIBLINEAR's SVM [16], and Batanović et al.'s implementation of NBSVM for WEKA[7] [2]. As suggested by Wang and Manning [15] we employ the L2 regularization and loss function for SVM and NBSVM. To ensure high test result replicability [17], [18] we evaluate using the same 10-run-average of 10-fold cross-validation as in [2]. The SVM and NBSVM hyperparameters are also optimized as in [2], through nested cross-validation.

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 [2], with two exceptions. Firstly, instead of the default WEKA tokenizer used in previous classifier evaluations, we employ the tokenizer for Serbian included in the ReLDI (*Regional Linguistic Data Initiative*) project repository[8] [19], [20] 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 [2], [15]. For the SVM we keep the token count features as they work better with classical discriminative algorithms [2]. We view the results obtained by utilizing these settings, without applying any morphological normalization, as a baseline. A paired corrected resampled *t*-test [18] was used to statistically compare the results of the morphologically normalized models with the baseline.

A. Unigram model

We first evaluate the lemmatizers on a unigram model, where 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 [2], we also reevaluate the stemmers for Serbian. The figures for the binary and the multiclass classification are given in Tables 1 and 2.

Table 1: Classifier CV accuracies on SerbMR-2C: Positive/Negative.

Morphological normalization	MNB	SVM	NBSVM
No normalization	80.18	82.00	83.50
Stemmers			
Kešelj and Šipka – optimal	**81.32**	83.32	84.01
Kešelj and Šipka – greedy	80.45	83.16	83.73
Milošević	81.04	**83.49**	**84.74**
Ljubešić and 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**

[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://vukbatanovic.github.io/NBSVM-Weka/

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

TABLE 2: CLASSIFIER CV ACCURACIES ON SERBMR-3C: POSITIVE/NEUTRAL/NEGATIVE.

Morphological normalization	MNB	SVM
No normalization	58.22	60.86
Stemmers		
Kešelj and Šipka – optimal	58.65	61.68
Kešelj and Šipka – greedy	58.04	60.92
Milošević	58.20	61.86
Ljubešić and 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**

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 [2]), 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 the vocabulary size and data sparsity. Derivationally related words most often do not express differing sentiments, so few classification errors are incurred due to this effect. Lemmatizers, on the other hand, focus solely on inflectional morphology, which limits their capabilities in vocabulary reduction.

Table 3 confirms this intuition by showing the vocabulary sizes of the SerbMR dataset[9] 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.

TABLE 3: VOCABULARY SIZE AS A FUNCTION OF MORPHOLOGICAL NORMALIZATION.

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

B. *Adding bigrams and trigrams*

Next, we explore how the addition of bigram and trigram features into the 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 simple 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 methods 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* denotes unigram, *B* bigram, and *T* trigram features. 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 very similar results to those obtained in [2].

TABLE 4: 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 and Pandžić	84.19	**86.11****	86.01**
(L) Ljubešić et al.	84.20	85.88*	85.84*

TABLE 5: 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 and Pandžić	62.05	**63.02***	62.42
(L) Ljubešić et al.	61.97	62.02	61.50

[9] The vocabulary size of the non-normalized dataset differs between this paper and [2] due to the use of different tokenization procedures.

TABLE 6: EXECUTION TIMES OF MORPHOLOGICAL NORMALIZERS.

Morphological normalization	Approximate execution time on SerbMR-3C	
Stemmers		
Kešelj and Šipka – optimal	~5s	
Kešelj and Šipka – greedy	~5s	
Milošević	~7s	
Ljubešić and 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)	

C. *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 6 contains the approximate execution times, since the exact figures slightly differ from one run to another. Most lemmatizers are several orders of magnitude slower than stemmers, and using them also requires taking into account the time to do POS tagging. The CST lemmatizer, used by Agić et al., is the only one comparable to stemmers with regard to speed, and it relies on the similarly fast *HunPos* tagger [21].

V. 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 using higher order n-gram models.

Our findings should make it easier to increase classifier performance levels 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 aim to extend our comparative 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] 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.

[3] 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.

[4] N. Milošević, “Stemmer for Serbian language.” arXiv 1209.4471, 2012.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.