Reliable Baselines for Sentiment Analysis in Resource-Limited Languages: The Serbian Movie Review Dataset

Vuk Batanović, Boško Nikolić, Milan Milosavljević
School of Electrical Engineering, University of Belgrade
Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia
E-mail: bv115045p@student.etf.bg.ac.rs, nbosko@etf.bg.ac.rs, mmilan@etf.bg.ac.rs

Abstract

Collecting data for sentiment analysis in resource-limited languages carries a significant risk of sample selection bias, since the small quantities of available data are most likely not representative of the whole population. Ignoring this bias leads to less robust machine learning classifiers and less reliable evaluation results. In this paper we present a dataset balancing algorithm that minimizes the sample selection bias by eliminating irrelevant systematic differences between the sentiment classes. We prove its superiority over the random sampling method and we use it to create the Serbian movie review dataset – SerbMR – the first balanced and topically uniform sentiment analysis dataset in Serbian. In addition, we propose an incremental way of finding the optimal combination of simple text processing options and machine learning features for sentiment classification. Several popular classifiers are used in conjunction with this evaluation approach in order to establish strong but reliable baselines for sentiment analysis in Serbian.

Keywords: sentiment classification, sample selection bias, dataset balancing

1. Introduction

Research on sentiment analysis has been copious, but so far mostly focused on English, a language with abundant resources. Relatively little attention has been paid to the specificities of sentiment analysis in languages in which resources are hard to find, or to the issues which only become pronounced under such conditions.

Sentiment classification is commonly benchmarked on the movie review domain, and most reference datasets in English are balanced (Pang, Lee, & Vaithyanathan, 2002; Pang & Lee, 2004; Maas et al., 2011). Hence, in order to explore how classifiers behave across different languages similarly sized balanced datasets are required.

Collecting data for sentiment analysis in a resource-limited setting requires making the most out of all available data sources. These can vary wildly in terms of size, class distribution, and various linguistic properties. Thus, the collected data is likely to be imbalanced and to suffer from high sample selection bias, as it is not representative of the whole population (Zadrozny, 2004; Bareinboim, Tian, & Pearl, 2014).

A common approach to constructing balanced datasets out of an imbalanced starting collection is random sampling. This technique works well if the initial data is free from sample selection bias (Van Hulse, Khoshgoftaar, & Napolitano, 2007). If it is not, however, it allows the bias to remain in the final balanced dataset.

Training and evaluating machine learning classifiers on such a dataset makes the classifiers less robust and the evaluation less reliable. Cross-validation performances are inflated, since the classifiers learn to rely on specious patterns in the dataset which are wholly incidental. Conversely, performances on independent test data are diminished, since test data does not necessarily suffer from the same bias as the training dataset.

We propose an alternative dataset balancing algorithm which attempts to minimize the sample selection bias. It does so by trying to eliminate any systematic differences between the samples placed in different classes except for those pertaining to their sentiment. By running this algorithm on an imbalanced collection of movie reviews in Serbian, we have created the Serbian movie review dataset – SerbMR – the first balanced, topically uniform sentiment analysis dataset in Serbian[1].

Additionally, we present an incremental way of choosing the optimal text processing options and machine learning features for sentiment classification. We analyze the impact of each step in the incremental approach and we evaluate its effectiveness on several commonly used machine learning algorithms, in both the binary and multiclass classification settings, in order to create a strong and reliable baseline for future comparisons.

This paper is structured as follows: in Section 2 we give an overview of the related work. Section 3 describes the dataset construction. The evaluation procedure and results are discussed in Section 4, while Section 5 contains our conclusions and some directions of future work.

2. Related Work

To the best of our knowledge, there is no previous work on reducing sample selection bias in the construction of balanced sentiment analysis datasets. Mountassir, Benbrahim, & Berrada (2012) discuss three alternatives to random undersampling in sentiment classification and evaluate them on datasets in English and Arabic, but do not consider the problem of sample selection bias. Blitzer, Dredze, & Pereira (2007) and Xia et al. (2013) tackle the closely related issue of domain adaptation in sentiment classification, with Elming, Hovy, & Plank (2014) focusing on low-resourced languages. Zadrozny (2004) proposes a general correction method for sample selection bias, but it presupposes that an explicit model of selection probabilities can be constructed. Ren et al. (2008) present an approach which reduces sample selection bias by first

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

exposing it in the dataset structure through clustering, and then rebalancing the dataset. However, this method requires the availability of additional unlabeled data.

Despite the recent deep-learning advances in sentiment analysis (e.g. (Kim, 2014; Johnson & Zhang, 2015)), several papers (Wang & Manning, 2012; Kim & Zhang, 2014) have shown that strong classification baselines can be established by using uncomplicated weighting techniques in conjunction with basic learning algorithms. This finding is particularly important for resource-limited languages, where the amounts of data typically necessary for deep learning are non-existent.

The only previous work on sentiment analysis in Serbian is a proof-of-concept solution by Milošević (2012a) and a paper by Mladenović et al. (2015). Milošević explored the viability of using the Naïve Bayes algorithm to classify sentences in Serbian according to their sentiment. Mladenović et al. presented a sentiment classification method which combines the maximum entropy classifier with the information from several external resources, including a sentiment lexicon and a variant of the Serbian WordNet (Krstev et al., 2004). They trained and evaluated their system on three sentiment analysis datasets in Serbian, two of which belong to the news domain, and one to the movie review domain. However, their movie review dataset is highly imbalanced and contains relatively few negative samples (347 or ~15%). On the other hand, the contents of different sentiment classes in their news domain datasets were drawn from sources which systematically differ from each other not only with regard to their sentiment but also their topic, leading to high topic bias (Brooke & Hirst, 2011; Ferschke, Gurevych, & Rittberger, 2013). These datasets are hence problematic from the standpoint of sentiment classification, since topic bias produces similar effects on classifier performances as sample selection bias.

3. Dataset Construction

Constructing the Serbian movie review dataset consisted of two phases – the acquisition of a raw imbalanced data collection, and its use in the creation of a balanced dataset.

3.1 Resource Acquisition

The key issue in resource acquisition was finding sufficient quantities of labeled data for model training. Since there are no large movie review websites like IMDB in Serbian, it was necessary to combine reviews gathered from several smaller sites, including blogs and local culture news portals. This necessity was compounded by the fact that most online reviews are positive, making negative reviews the critical resource. Sites with less than 100 reviews were dismissed as unacceptably small since they always contained very few examples of negative reviews. In order to minimize topic bias in the dataset we also avoided websites focused on a single movie genre. To reduce data sparsity only those sites whose contents are predominantly written in the Ekavian pronunciation were taken into consideration. Lastly, we discarded a few websites with prohibitive copyright policies. These selection criteria resulted in eight websites being accepted as sources for the movie review dataset.

Most of the accepted source sites use a 1–10 scoring scale, so we adopted it as the standard. Two websites (yc.rs and happynovisad.com) use a 1–5 scoring system, which is easily translated to 1–10 by multiplying each score by two. In such cases, a plus / minus next to the original score was treated as an increment / decrement of the translated score. Pluses / minuses in the 1–10 scoring systems were ignored and X.5 scores were rounded down to X. In a few rare cases where a zero score was given, it was translated into a score of one. Finally, review scores 1–4 were considered negative, 5–6 neutral, and 7–10 positive.

Tables 1 and 2 present an overview of the collected imbalanced data. Positive reviews outnumber the negative and the neutral ones on all websites, often considerably so. Positive reviews are also noticeably longer on average.

Source website / Review count	Neg	Neu	Pos	All
2kokice.com	87	242	385	714
filmskerecenzije.com	23	28	335	386
filmskihitovi.blogspot.com	120	247	304	671
happynovisad.com	32	34	98	164
kakavfilm.com	86	201	639	926
mislitemojomglavom.blogspot.com	246	192	265	703
popboks.com	182	265	476	923
yc.rs	65	69	104	238
Total count	841	1278	2606	4725
Average length (in tokens)	468	467	529	501

Table 1: The distribution of the collected movie reviews in Serbian according to their source website.

Review score		Review count
Negative	1	113
	2	154
	3	206
	4	368
Neutral	5	530
	6	748
Positive	7	785
	8	951
	9	489
	10	381

Table 2: The distribution of the collected movie reviews in Serbian according to their score.

3.2 Dataset Balancing

We tackle the problem of creating a balanced dataset as one of finding the best neutral and positive pairing for each available negative review. To do so, our algorithm takes into account the following factors:

- Review scores – in order to ensure that the opposing sentiment classes contain equally strong sentiment terms, negative reviews are paired only to the positive ones with an inverse polarity score (e.g. a 1 with a 10, a 4 with a 7). Omitting this criterion could make three-

class classification (positive / neutral / negative) harder to learn, since the more weakly expressed polar class would become closer to the neutral subset than its opposing class. Keeping the neutral class equally distant to both of the polar classes also requires that the neutral subset be composed of an equal number of reviews with the score 5 and those with the score 6.

- Review sources – different source sites use various styles and levels of formality in writing which have to be balanced out between the sentiment classes. Ignoring these linguistic traits would lead to sample selection bias, with a specific vocabulary or a certain register becoming strongly but mistakenly correlated with one class and weakly with the other(s).
- Review lengths – the difference in lengths (i.e. token counts) between the paired reviews and between the sentiment classes in general should be minimal. If a review is significantly longer than its pair then the balance of the aforementioned linguistic traits becomes endangered. Moreover, if one of the polar classes is much larger than the other, it will certainly contain more neutral words. A classifier trained on such a dataset will therefore erroneously learn to treat objectively neutral words as if they are somewhat indicative of a certain polarity.

For each negative review Neg the algorithm forms a positive-pair candidate list that consists of inversely-scored positive reviews Pos from the same source site where $|length(Pos) - length(Neg)| < diff$, with the starting value $diff_{start} = 100$. If no candidates exist the source and the length criteria are slowly relaxed until candidates appear. This relaxation is cyclical – the first step is to accept reviews from the other source sites and the second is to increase $diff$ in steps of 50, but doing so retriggers the same source site requirement. The source criterion is the first one to be relaxed, since increasing the length imbalance damages the fair distribution of both neutral words and various linguistic features across the sentiment classes. The score criterion is kept fixed during the whole process, because of the strong imbalance in the number of reviews from each individual score category.

After the candidate lists have been formed, negative reviews are paired with positive ones in increasing order of the number of candidates they have. This ordering not only minimizes the chances of having to repeat a candidate search (due to all existing candidates being already assigned to previously processed negative reviews), but also improves the global selection quality, since each repeated search requires either the removal of the source criterion, an increase of the $diff$ value, or both. The best positive candidate for a given negative review is the one where the length differential is the smallest. If there are several equally good candidates, the algorithm tries to find one where the sign of the length differential between the reviews is such that it reduces the global difference in token counts between the classes.

Negative reviews are paired with the neutral ones in a similar fashion. The only difference is that review scores do not play a role in the formation of candidate lists, but instead in the choice of the best neutral candidates. The neutral subset ought to be split evenly between scores 5 and 6, so the algorithm keeps a global tally of these two score groups during candidate selection. For each negative review it picks the best candidate among those that belong to the group currently in the minority. If no reviews exist in the minority score list, the algorithm accepts the best majority score candidate. A temporary score distribution imbalance is thus allowed, as it can still be corrected in the following negative-neutral pairings. Candidate scores become irrelevant in the moments when both score groups are equally numerous.

Table 3 shows a comparison between the final, balanced SerbMR and Pang & Lee's datasets in English (Pang et al., 2002; Pang & Lee, 2004). Although it has more reviews than the English MR 1.0 dataset, the polar portion of SerbMR is actually smaller by 200 000 tokens[2]. Moreover, due to the morphological complexity of Serbian, the vocabulary in SerbMR is about 2.5 times larger than the one in English MR 1.0, and the average number of occurrences of each word in the dataset ($|N|/|V|$) is around three times lower than in English. Lastly, the sentiment classes in SerbMR are very balanced with regard to their token count, thanks to our balancing algorithm, while in the English datasets, produced by random sampling, the positive class is ~10% larger.

Sentiment class	Counts/Size	SerbMR	English MR 1.0	English MR 2.0
Negative	Reviews	841	700	1000
	Tokens ($\|N\|$)	393K	470K	668K
	Average length	468	672	668
	Vocabulary ($\|V\|$)	73K	30K	35K
	$\|N\|/\|V\|$	5	16	19
Neutral	Reviews	841	/	/
	Tokens ($\|N\|$)	394K	/	/
	Average length	469	/	/
	Vocabulary ($\|V\|$)	70K	/	/
	$\|N\|/\|V\|$	6	/	/
Positive	Reviews	841	700	1000
	Tokens ($\|N\|$)	398K	522K	743K
	Average length	473	745	743
	Vocabulary ($\|V\|$)	72K	32K	37K
	$\|N\|/\|V\|$	6	16	20
Negative + Positive	Reviews	1682	1400	2000
	Tokens ($\|N\|$)	791K	992K	1411K
	Average length	470	709	706
	Vocabulary ($\|V\|$)	116K	44K	51K
	$\|N\|/\|V\|$	7	23	28
All	Reviews	2523	1400	2000
	Tokens ($\|N\|$)	1185K	992K	1411K
	Average length	470	709	706
	Vocabulary ($\|V\|$)	148K	44K	51K
	$\|N\|/\|V\|$	8	23	28

Table 3: A comparison between SerbMR and the existing English datasets of similar size and domain.

[2] The token counts do not include the punctuation marks.

4. Evaluation

We present two separate evaluation tracks in this paper. To begin with, we examine the performance of the proposed dataset balancing algorithm. Afterwards, we move on to the task of sentiment analysis – we introduce our incremental evaluation approach and review the results obtained by using it in conjunction with a set of machine learning classifiers.

Evaluation is performed within the WEKA workbench (Hall et al., 2009), in both the binary (SerbMR-2C: only positive and negative reviews) and the full three-class (SerbMR-3C) setting. We train three classifiers often used as sentiment analysis baselines: WEKA's implementation of Multinomial Naïve Bayes (MNB) and LIBLINEAR (Fan et al., 2008) versions of Logistic Regression (LR), and Support Vector Machines (SVM). Per (Wang & Manning, 2012), we use the L2 regularization for LR and SVM and the L2 loss function for SVM. In order to ensure high test replicability, classifier accuracies are obtained by averaging 10 runs of 10-fold cross-validation (Bouckaert, 2003; Bouckaert & Frank, 2004). The hyperparameters of LR and SVM are optimized through nested cross-validation via the *MultiSearch* [3] WEKA package in the following ranges:

- Cost $C \in [10^{-3} - 10^{3}]$
- Bias $b \in \{0, 1\}$
- The epsilon tolerance $eps \in [10^{-3} - 10^{1}]$
- The use of the LIBLINEAR normalization: On / Off

4.1. Dataset Balancing

Dataset balancing algorithms are evaluated with the default initial WEKA settings (1000 most frequent binary unigram features per class). We generate 10 different balanced subsets, of the same size as SerbMR, by including all the negative reviews from the corpus of collected reviews and randomly sampling 841 neutral and positive ones. We train and evaluate classifiers using these datasets, average their performances, and compare them to the accuracies achieved by using SerbMR.

Table 4 shows classifiers reaching higher cross-validation accuracies on randomly sampled datasets, which is an expected effect of the sample selection bias still present in them. Random undersampling cannot remove the non-sentiment-related differences between the classes, making cross-validation, in effect, easier, as classifiers can be aided by irrelevant regularities that remain in the data.

For instance, informally written reviews are significantly more common than formal ones on three of the accepted source sites (2kokice.com, filmskihitovi.blogspot.com, and mislitemojomglavom.blogspot.com), while on the other websites reviews generally tend to be written in a semi-formal to formal register. If we consider the distribution of collected movie reviews according to their source website, it becomes clear that the percentage of informal reviews in the negative category is much higher than in the positive or the neutral one. Random undersampling cannot remedy this and classifiers trained on randomly sampled datasets will thus learn to associate informal expressions much more strongly with the negative class. However, such patterns are not general and learning them makes the classifiers less robust. Unlike random undersampling, our dataset balancing algorithm is able to minimize these specious regularities by taking review sources, lengths, and scores into consideration.

This effect becomes evident when we test the classifiers on a separate balanced test set of short movie comments, which was manually annotated. Sentiment analysis of short texts is generally much harder than document-level classification, leading to significant performance drops. Nevertheless, Table 5 demonstrates that the classifiers trained on SerbMR do better on independent test data than those trained on randomly sampled datasets.

Dataset	MNB	LR	SVM
Positive / Negative			
Randomly sampled subsets	**76.78**	**78.77**	**78.83**
SerbMR	75.29	75.75	75.98
Positive / Neutral / Negative			
Randomly sampled subsets	**56.42**	**58.28**	**58.41**
SerbMR	54.72	55.14	55.49

Table 4: Classifier CV accuracies on randomly sampled subsets and the SerbMR dataset.

Dataset	MNB	LR	SVM
Positive / Negative			
Randomly sampled subsets	59.79	59.75	59.55
SerbMR	**61.56**	**61.33**	**62.18**
Positive / Neutral / Negative			
Randomly sampled subsets	46.27	43.95	43.66
SerbMR	**47.18**	**45.82**	**45.77**

Table 5: Test set accuracies of classifiers trained on randomly sampled subsets or the SerbMR dataset.

4.2. Sentiment Analysis

We find the optimal baseline settings by incrementally experimenting on SerbMR with different text processing options and machine learning features. We then consider classifier performances under similar settings in English and Serbian, and try to apply the optimal options to NBSVM (Wang & Manning, 2012), a combination of NB and SVM designed for topic and sentiment classification.

To avoid overfitting to a single classifier, we accept only the changes whose overall impact is positive, even if it comes at the cost of a mild drop in the performance of a particular classifier. The only exception is the adoption of a minimal n-gram frequency in the binary classification setting, which slightly reduces classifier accuracies, but drastically improves their speed. Options approved and used in subsequent experiments are marked in boldface in the result tables, while the rejected ones are crossed over. Boldface also highlights the figures that are the best in a group of options and those better than the results of options currently accepted in the incremental evaluation.

[3] http://github.com/fracpete/multisearch-weka-package/

4.2.1. Dealing with Negation

Pang et al. (2002) proposed a simple technique for dealing with negations in which a prefix is added to all the words between a negation and the nearest following punctuation mark. We first examine whether limiting the scope of this technique to fewer words can be beneficial, and we do so with the default initial WEKA settings. The results are shown in Table 6. It can be concluded that marking only the first one or two words after a negation outperforms the other variants.

Settings	MNB	LR	SVM
SerbMR-2C: Positive / Negative			
Initial default settings	75.29	75.75	75.98
Mark words with negation prefixes after a negation			
Until the first punctuation mark	74.99	75.06	75.28
1 word after a negation word	75.62	76.31	**76.64**
2 words after a negation word	**75.67**	**76.33**	76.45
3 words after a negation word	75.52	76.15	76.25
5 words after a negation word	75	75.45	75.58
10 words after a negation word	75.29	75.28	75.57
SerbMR-3C: Positive / Neutral / Negative			
Initial default settings	54.72	55.14	55.49
Mark words with negation prefixes after a negation			
Until the first punctuation mark	54.75	54.84	55.26
1 word after a negation word	55.62	**55.89**	56.1
2 words after a negation word	**55.66**	55.81	**56.13**
3 words after a negation word	55.37	55.71	55.93
5 words after a negation word	55.08	55.32	55.51
10 words after a negation word	55.1	55.29	55.58

Table 6: Classifier CV accuracies on the SerbMR dataset when applying different negation-marking techniques.

4.2.2. Features

The best configuration of machine learning features and their types is determined in four stages. We first find the optimal *feature count* by experimenting with the WEKA settings which limit the number of features according to their frequency in each class or the whole dataset. Lowercasing all tokens is also considered. We then evaluate various changes to *feature values*, including the use of token count features instead of the binarized ones, as well as the effects of weighting strategies such as TF/IDF and length normalization.

Afterwards, we focus on *stemming*. We have separated stemming from the other methods which influence the feature count since stemmers are language-specific tools whose impact is dependent on the nature of the language in question. For instance, stemming is rarely used for sentiment analysis in English, but we suspect that in morphologically complex and resource-limited languages like Serbian it may aid the classifiers. We experiment with four stemming algorithms: the optimal and the greedy stemmers of Kešelj & Šipka (2008), the improvement of the greedy algorithm proposed by Milošević (2012b), and

SerbMR-2C: Positive / Negative			
Settings	MNB	LR	SVM
Mark 1 word after a negation word	75.62	76.31	76.64
Feature count			
Number of (most frequent) n-grams used			
Increased to 10 000 per class	**80.06**	79.22	79.19
Increased to 100 000 per class / All unigrams	79.93	**79.44**	**79.58**
Minimal n-gram frequency			
Increased to 2 per class	79.54	**79.45**	**79.75**
Increased to 3 per class	79.8	79.3	79.35
Increased to 4 per class	**80.06**	79.22	79.19
Increased to 5 per class	79.8	79.02	79.1
***Number of n-grams used* and *Minimal n-gram frequency* refer to the entire dataset**	79.76	**79.31**	**79.44**
Lowercase tokens	**80.21**	**79.66**	**79.53**
Feature values			
Non-binary (token count) features	79.51	**81.94**	**81.73**
TF normalization	**80.31**	**82.29**	**82.04**
~~IDF normalization~~	76.86	**82.63**	**82.33**
~~Document length normalization~~	80.2	81.71	81.57
Stemming			
Kešelj & Šipka – optimal	80.98	83.67	83.35
Kešelj & Šipka – greedy	80.3	83.28	83.08
Milošević	80.74	83.75	83.69
Ljubešić & Pandžić	**81.19**	**84.02**	**83.95**
Bigram & trigram features			
Unigrams + bigrams	83.26	84.07	84.02
Unigrams + bigrams + trigrams	**83.66**	**84.44**	**84.25**

Table 7: Classifier CV accuracies on the SerbMR-2C dataset obtained in the incremental evaluation.

a stemmer for Croatian, a language closely related to Serbian, by Ljubešić & Pandžić[4], which is a refinement of the algorithm presented in (Ljubešić, Boras, & Kubelka, 2007). Each stemmer was originally coded in a different programming language, so we have re-implemented them all in a unified framework as a WEKA package[5].

Finally, we evaluate the inclusion of *bigram and trigram features*. The results for binary classification are shown in Table 7, while Table 8 contains the three-class figures.

Despite some differences in the optimal options for the binary and the three-class setting, several consistencies can be observed. We reaffirm the conclusion of Pang et al. (2002) and Wang & Manning (2012) that binarized features work better for MNB, but unlike Pang et al. we find the discriminative classifiers boosted by non-binary features. Lowercasing tokens and TF normalization bring an improvement across the board, whereas including IDF

[4] http://nlp.ffzg.hr/resources/tools/stemmer-for-croatian/

[5] Available at: http://vukbatanovic.github.io/SCStemmers/

SerbMR-3C: Positive / Neutral / Negative			
Settings	MNB	LR	SVM
Mark 2 words after a negation word	55.66	55.81	56.13
Feature count			
Number of (most frequent) n-grams used			
Increased to 10 000 per class	**57.53**	57.88	57.89
Increased to 100 000 per class / All unigrams	56.63	**57.99**	**58.22**
~~Minimal n-gram frequency~~			
Increased to 2 / 3 / 4 per class	57.53	57.88	57.89
Increased to 5 per class	**57.56**	57.59	57.69
Increased to 6 per class	57.45	57.59	57.75
Increased to 7 per class	56.96	57.44	57.67
~~*Number of n-grams used* and *Minimal n-gram frequency* refer to the entire~~ dataset	56.88	57.53	57.69
Lowercase tokens	**58.10**	**58.43**	**58.62**
Feature values			
Non-binary (token count) features	57.94	**59.05**	**59.06**
TF normalization	**58.52**	**60.59**	**60.57**
~~IDF normalization~~	54.16	60.39	59.07
~~Document length normalization~~	58.24	59.62	59.54
Stemming			
Kešelj & Šipka – optimal	57.91	61.69	61.78
Kešelj & Šipka – greedy	57.33	60.91	60.83
Milošević	57.91	62.09	62.02
Ljubešić & Pandžić	**58.87**	**62.26**	**62.14**
Bigram & trigram features			
Unigrams + bigrams	60.75	**63.15**	**62.85**
Unigrams + bigrams + trigrams	**61.01**	62.66	62.57

Table 8: Classifier CV accuracies on the SerbMR-3C dataset obtained in the incremental evaluation.

(i.e. using TF-IDF) can sometimes help LR and SVM, but greatly lowers the accuracy of MNB. Length normalization is not found to be useful, but this might be a side effect of our dataset balancing algorithm.

As expected, we find that stemming can markedly improve classifier accuracies in morphologically rich but resource-limited languages. By normalizing different forms of a same word, stemmers lower the vocabulary size of each sentiment class and of the entire SerbMR by ~30–35% and increase the average number of occurrences of each word ($|N|/|V|$) by ~50%. This reduces data sparsity and allows the classifiers to better model the impact of each word. The effects of different stemmers on classifier accuracies are far from the same, yet they all shrink the vocabulary to a similar extent. Thus, these differences must be caused by the quality of the stemming rules used in each algorithm. While our conclusions might not generalize to other NLP tasks, the results show that the stemmer of Ljubešić & Pandžić is the best one for sentiment classification of documents in Serbian.

Finally, we reiterate the findings of many previous papers that combining unigram and bigram features outperforms purely unigram models. Adding trigram features helped us only in the binary classification setting, where the differences between the classes are more pronounced, making trigram features less likely to introduce noise.

Positive / Negative				
Dataset / Settings	Minimal n-gram frequency	MNB	LR	SVM
Unigram features				
English non-binary	4	78.7	/	72.8
English binary	4	81	80.4	82.9
Serbian non-binary[1]	4	78.96	80.89	80.52
Serbian binary[2]	4	79.48	78.57	79.57
Serb optimal	4	**80.82**	**83.30 *[1] **[2]**	**82.66 *[2]**
Bigram features				
English binary	7	77.3	77.4	77.1
Serbian binary	7	74.57	71.98	72.73
Serbian optimal	7	**76.5**	**74.81***	**74.77**
Unigram + bigram features				
English binary	4 – unigram 7 – bigram	80.6	80.8	82.7
Serbian binary	4	81.64	79.89	80.56
Serbian optimal	4	**82.68**	**83.83****	**83.51***
Serbian binary	7	80.61	79.28	79.45
Serbian optimal	7	**82.37**	**83.33****	**82.95****

Table 9: An overview of classifier CV accuracies on English and Serbian (English MR 1.0 and SerbMR-2C).

4.2.3. Classifier Performances Across Languages

In order to make a fair assessment about how classifiers perform across different languages – English and Serbian – the amount of training data given to them has to be taken into account. Conveniently, a three-fold cross-validation on the English MR 1.0 dataset yields a similar number of tokens for training as a five-fold cross-validation on the polar portion of SerbMR (661K vs 633K). Therefore, we average 20 runs of five-fold cross-validation on SerbMR-2C and consider two types of settings. In one we emulate the text processing options and machine learning features used by Pang et al. (2002), and in the other we employ the optimal settings found through our incremental evaluation. Table 9 contains an overview of classifier accuracies, with the English-language figures taken from (Pang et al., 2002). The results of the two types of settings for Serbian are evaluated with a paired corrected resampled *t*-test (Bouckaert & Frank, 2004). The differences found statistically significant at the 0.05 / 0.01 level are marked with * / **.

Pang et al. did not optimize classifier hyperparameters, but the accuracies achieved on English MR 1.0 are mostly similar to or even better than the ones obtained on SerbMR by using the same settings *with* hyperparameter optimization[6]. This discrepancy is a consequence of the greater morphological complexity of Serbian. Still, our optimal settings lead to results which are oftentimes significantly better than the ones with the default settings, particularly in the case of discriminative classifiers.

4.2.4. NBSVM

As a final point, we explore whether applying the optimal settings to NBSVM (Wang & Manning, 2012), a simple but strong baseline algorithm designed for topic and sentiment classification, might improve the results even further. NBSVM was originally devised as a binary classifier that combines the Multinomial Naïve Bayes algorithm with Support Vector Machines. It does so through the element-wise multiplication of the SVM feature vector f by the positive class/negative class ratio vector r of MNB log-counts:

$$p = \alpha + \sum_{i:Class(i)=Pos} f^{(i)}$$
$$q = \alpha + \sum_{i:Class(i)=Neg} f^{(i)}$$
$$r = \log\left(\frac{p/\|p\|_1}{q/\|q\|_1}\right)$$
$$\tilde{f}^{(k)} = r \circ f^{(k)}$$

where p and q are the count vectors for the positive and the negative class, and α is the smoothing parameter. The final feature vector $\tilde{f}$is used as input to a standard SVM classifier. The original algorithm utilizes binarized features but it can work with non-binary ones as well.

We extend NBSVM to support one-vs-all multiclass classification – the model for N classes consists of N binary classifiers and N separate ratio vectors r. For each class C we calculate:

$$p = \alpha + \sum_{i:Class(i)=C} f^{(i)}$$
$$not\_p = \alpha + \sum_{i:Class(i)\neq C} f^{(i)}$$
$$r_C = \log\left(\frac{p/\|p\|_1}{not\_p/\|not\_p\|_1}\right)$$
$$\tilde{f_C}^{(k)} = r_C \circ f^{(k)}$$

where $\tilde{f_C}$ is the feature vector used as input for the binary classifier for class C. In essence, we separately treat each class as *positive* and all the other classes combined as *negative*. The final classification is performed in the usual one-vs-all manner – we choose the class whose binary SVM classifies the given sample with the greatest margin. We implemented this multiclass version of NBSVM as a WEKA package[7] that relies on the LIBLINEAR library.

As Wang & Manning (2012) do, we also apply the interpolation between MNB and SVM to determine the final model weights w':

$$w' = (1-\beta)\frac{\|w\|_1}{|V|} + \beta w$$

where w is the weight vector obtained by training the SVM using the $\tilde{f}$ feature vector, $\|w\|_1/|V|$ is its mean magnitude, and β is the interpolation parameter. This technique is naturally extendable to multiclass settings – the weights of each binary SVM are interpolated separately but with the same global β value.

We again employ the L2 regularization and loss function for SVM. Nested cross-validation is used to optimize both the four previously described SVM hyperparameters, in the same ranges as before, and the NBSVM interpolation parameter, with the recommended values $\beta \in \{0.25, 0.5\}$. As is usual, we set $\alpha = 1$. Accuracies are again obtained by averaging 10 runs of 10-fold cross-validation.

As the starting point in the evaluation of NBSVM we take the optimal feature count settings determined through our incremental approach. We then explore the impact of utilizing all optimal settings (i.e. moving to token count features, TF-normalization and stemming), in conjunction with the inclusion of bigram and trigram features. Lastly, since MNB works better with binarized features, unlike SVM where the reverse is the case, we consider if the change within the optimal settings back to binary features can yield any improvements. The results for binary and three-class classification are shown in Table 10.

Evaluation shows that our optimal settings prove effective on NBSVM in all configurations. We can conclude that, although NBSVM is already built on a special weighting scheme, additional weighting in the form of TF normalization, coupled with stemming, improves the

NBSVM		
Settings	SerbMR-2C	SerbMR-3C
Unigram features		
Optimal feature counts	83.26	59.93
Full optimal settings	84.24	61.05
Full optimal settings + binary features	**84.33**	**61.29**
Unigram + bigram features		
Optimal feature counts	84.05	61.01
Full optimal settings	**85.45**	61.69
Full optimal settings + binary features	85.36	**62.69**
Unigram + bigram + trigram features		
Optimal feature counts	84.05	60.52
Full optimal settings	85.52	61.54
Full optimal settings + binary features	**85.55**	**62.24**

Table 10: NBSVM CV accuracies on the SerbMR-2C and SerbMR-3C datasets.

[6] It was not possible to just replicate their hyperparameter settings in our work due to the use of different machine learning libraries with distinct parameters.

[7] Available at: http://vukbatanovic.github.io/NBSVM-Weka/

classifier's performance. The results on SerbMR-2C demonstrate that in the binary classification task the difference between binarized and non-binary features is negligible, particularly if higher-order n-gram features are utilized. However, in the three-class setting binarized features have a clear upper hand.

Similarly to the previously considered algorithms, NBSVM performs best on the binary classification task when using a combination of unigram, bigram and trigram features, whereas excluding the trigrams consistently helps in the three-class setting. NBSVM performs noticeably better on binary classification than the basic classifiers and raises the maximum accuracy on SerbMR-2C by ~1%. The same cannot be said of three-class classification where, despite the effects of the optimal settings and binarized features, NBSVM fails to surpass or even catch up to them.

We believe that this behavior is caused by the nature of NBSVM ratio vectors. In the binary setting, where classes are clearly separated, these vectors aid the classifier by increasing the importance of features which appear frequently in one class but rarely in the other. Such features are usually good indicators of sentiment since positive words seldom occur in negative reviews and vice versa. Simultaneously, the ratio vectors lower the significance of features which occur with similar frequencies in both classes and which are, thus, uninformative for sentiment analysis.

The problem for NBSVM is that in the SerbMR-3C dataset the sentiment classes are not clearly separated, since the neutral class does not contain some objective information but rather a mixture of positive and negative sentiments. Therefore, there are not many terms which often appear in the neutral class and rarely in the others, since the sentiment of the neutral class is less conveyed by some particular *neutral* expressions and more by a blend of *positive* and *negative* ones. The NBSVM algorithm is incapable of modeling relations of this sort, making its effectiveness limited in situations of this kind.

The use of MNB log-count ratios can easily be extended to other discriminative classifiers – Mesnil et al. (2015) used logistic regression instead of SVM. We also experimented with replacing SVM with LR but consistently achieved near-identical results.

5. Conclusion

In this paper we have considered the issue of sample selection bias encountered during resource acquisition in resource-limited languages. We have found that a specifically-designed algorithm can surpass random undersampling in minimizing this problem on the task of building balanced sentiment datasets. By using this algorithm we have created the Serbian movie review dataset, the first balanced and topically uniform sentiment analysis dataset in Serbian. In addition, we have proposed an incremental evaluation procedure which allowed us to discover the optimal combination of simple text processing options and machine learning features for sentiment classification and to utilize them in order to establish strong baseline results. We have also been able to combine our findings with the NBSVM classifier for an even better performance on binary classification.

In the future we aim to make use of our dataset-building approach in other domains, including music, books, and product reviews, and to thereby increase the amount of data that can be used for sentiment analysis in Serbian. This could potentially enable the successful application of sentiment analysis models which require large amounts of training data. We are also interested in tackling the specificities of short-text processing in morphologically complex languages like Serbian.

6. Acknowledgements

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

7. Bibliographical References

Bareinboim, E., Tian, J., and Pearl, J. (2014). Recovering from Selection Bias in Causal and Statistical Inference. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014)*, Québec City, Québec, Canada, AAAI Press, pp. 2410–2416.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, Prague, Czech Republic, Association for Computational Linguistics, pp. 440–447.

Bouckaert, R. R. (2003). Choosing between two learning algorithms based on calibrated tests. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003)*, Washington DC, USA, AAAI, pp. 51–58.

Bouckaert, R. R., and Frank, E. (2004). Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms. In *Proceedings of the Eighth Pacific-Asia Conference (PAKDD 2004)*, Sydney, Australia, Springer Berlin Heidelberg, pp. 3–12.

Brooke, J., and Hirst, G. (2011). Native language detection with "cheap" learner corpora. In *Proceedings of the First Learner Corpus Research Conference (LCR 2011)*, Louvain-la-Neuve, Belgium, pp. 37–47.

Elming, J., Hovy, D., and Plank, B. (2014). Robust Cross-Domain Sentiment Analysis for Low-Resource Languages. In *Proceedings of the Fifth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA 2014)*, Baltimore, Maryland, USA, Association for Computational Linguistics, pp. 2–7.

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

Ferschke, O., Gurevych, I., and Rittberger, M. (2013). The Impact of Topic Bias on Quality Flaw Prediction in Wikipedia. In *Proceedings of the 51st Annual Meeting of*

the Association for Computational Linguistics (ACL 2013), Sofia, Bulgaria, Association for Computational Linguistics, pp. 721–730.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1), pp. 10–18.

Johnson, R., and Zhang, T. (2015). Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS 2015)*, Montreal, Canada.

Kešelj, V., and Šipka, D. (2008). A Suffix Subsumption-Based Approach to Building Stemmers and Lemmatizers for Highly Inflectional Languages with Sparse Resources. *INFOtheca*, 9(1-2), p. 23a–33a.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, Doha, Qatar, Association for Computational Linguistics, pp. 1746–1751.

Kim, Y., and Zhang, O. (2014). Credibility Adjusted Term Frequency: A Supervised Term Weighting Scheme for Sentiment Analysis and Text Classification. In *Proceedings of the Fifth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Baltimore, Maryland, USA, Association for Computational Linguistics, pp. 79–83.

Krstev, C., Pavlović-Lažetić, G., Vitas, D., and Obradović, I. (2004). Using Textual and Lexical Resources in Developing Serbian Wordnet. *Romanian Journal of Information Science and Technology*, 7(1-2), pp. 147–161.

Ljubešić, N., Boras, D., and Kubelka, O. (2007). 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, pp. 313–320.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, Portland, Oregon, USA, Association for Computational Linguistics, pp. 142–150.

Mesnil, G., Ranzato, M., Mikolov, T., and Bengio, Y. (2015). Ensemble of Generative and Discriminative Techniques for Sentiment Analysis of Movie Reviews. In *Proceedings of the ICLR 2015 Workshop Track*.

Milošević, N. (2012a). Mašinska analiza sentimenta rečenica na srpskom jeziku, *Master's Degree Thesis*, University of Belgrade, Belgrade, Serbia. [in Serbian]

Milošević, N. (2012b). Stemmer for Serbian language. arXiv 1209.4471.

Mladenović, M., Mitrović, J., Krstev, C., and Vitas, D. (2015). Hybrid sentiment analysis framework for a morphologically rich language. *Journal of Intelligent Information Systems*.

Mountassir, A., Benbrahim, H., and Berrada, I. (2012). An empirical study to address the problem of Unbalanced Data Sets in Sentiment Classification. In *Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2012)*, Seoul, South Korea, IEEE, pp. 3298–3303.

Pang, B., and Lee, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, Morristown, New Jersey, USA, Association for Computational Linguistics, Article No. 271.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, Philadelphia, Pennsylvania, USA, Association for Computational Linguistics, pp. 79–86.

Ren, J., Shi, X., Fan, W., and Yu, P. S. (2008). Type Independent Correction of Sample Selection Bias via Structural Discovery and Re-balancing. In *Proceedings of the 2008 SIAM International Conference on Data Mining (SDM 2008)*, Atlanta, Georgia, USA, SIAM, pp. 565–576.

Van Hulse, J., Khoshgoftaar, T. M., and Napolitano, A. (2007). Experimental Perspectives on Learning from Imbalanced Data. In *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, Corvallis, Oregon, USA, pp. 935–942.

Wang, S., and Manning, C. D. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, Jeju Island, South Korea, Association for Computational Linguistics, pp. 90–94.

Xia, R., Zong, C., Hu, X., and Cambria, E. (2013). Feature Ensemble Plus Sample Selection: Domain Adaptation for Sentiment Classification. *IEEE Intelligent Systems*, 28(3), pp. 10–18.

Zadrozny, B. (2004). Learning and Evaluating Classifiers under Sample Selection Bias. In *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, Banff, Alberta, Canada.