Discriminating between VERY similar languages among Twitter users

Nikola Ljubešić, Denis Kranjčić

Department of Information and Communication Sciences
Faculty of Humanities and Social Sciences
University of Zagreb
I. Lučića 3, HR-10000 Zagreb
{nljubesi,dkranjci}@ffzg.hr

Abstract

In this paper we tackle the problem of discriminating Twitter users by the language they tweet in, taking into account very similar South-Slavic languages, namely Bosnian, Croatian, Montenegrin and Serbian. We take the supervised machine learning approach by annotating a subset of 500 users from an existing Twitter collection by the language they primarily tweet in. We show that by using either words or character 6-grams as features, univariate feature selection, up to 10% of most significant features and a standard classifier, on the user level we reach accuracy of ∼97%.

Razlikovanje med ZELO podobnimi jeziki uporabnikov Twitterja

V prispevku raziskujemo problem ločevanja uporabnikov družabnega omrežja Twitter glede na to, v katerem jeziku tvitajo, pri čemer obravnavamo zelo podobne južnoslovanske jezike: bosanščino, hrvaščino, srbščino in črnogorščino. Uporabimo pristop z nadzorovanim strojnim učenjem, kjer označimo vsakega uporabnika iz že obstoječe podatkovne množice 500 uporabnikov z jezikom, v katerem tvita. Pokažemo, da z uporabo besed ali 6-gramov znakov kot značilk, univariantno izbiro značilk, do 10% najpomembnejših značilk in standardnega klasifikatorja dosežemo ∼97% točnost pravilne klasifikacije posameznega uporabnika.

1. Introduction

The problem of language identification, which was considered a solved task for some time now, has recently gained in popularity among researchers by identifying more complex problems such as discriminating between language varieties (very similar languages and dialects), identifying languages in multi-language documents, code-switching (alternating between two or more languages) and identifying language in very short documents (such as tweets).

In this paper we address the first and the last problem, namely discriminating between very similar languages in Twitter posts, with the restriction that we do not identify language on the tweet level, but the user level.

The four languages we focus on here, namely Bosnian, Croatian, Montenegrin and Serbian, belong to the South Slavic group of languages and are all very similar to each other.

All the languages, except Montenegrin, use the same phonemic inventory, and they are all based on the write-as-you-speak principle. Croatian is slightly different in this respect, because it does not transcribe foreign words and proper nouns, as the others do. Moreover, due to the fairly recent standardization of Montenegrin, its additional phonemes are extremely rarely represented in writing, especially in informal usage. The Serbian language is the only one where both Ekavian and Ijekavian pronunciation and writing are standardized and widely used, while all the other languages use Ijekavian variants as a standard. The languages share a great deal of the same vocabulary, and some words differ only in a single phoneme, because of phonological, morphological and etymological circumstances. There are some grammatical differences regarding phonology, morphology and syntax, but they are arguably scarce and they barely influence mutual intelligibility. The distinction between the four languages is based on the grounds of establishing a national identity, rather than on prominently different linguistic features.

2. Related work

One of the first studies incorporating similar languages in a language identification setting was that of Padró and Padró (2004) who, among others, discriminate between Spanish and Catalan with an accuracy of up to 99% by using second order character-level Markov models. In (Ranaivo-Malancon, 2006) a semi-supervised model is presented to distinguish between Indonesian and Malay by using frequency and rank of character trigrams derived from the most frequent words in each language, lists of exclusive words, and the format of numbers. Huang and Lee (2008) use a bag-of-words approach to classify Chinese texts from the mainland and Taiwan with results of up to 92% accuracy. Zampieri and Gebre (2012) propose a log-likelihood estimation method along with Laplace smoothing to identify two varieties of Portuguese (Brazilian and European) obtaining 99.5% accuracy.

In the first attempt at discriminating between the two most distant out of the four languages of interest, namely Croatian and Serbian, Ljubešić et al. (2007)have shown that by using a second-order character Markov chain and a list of forbidden words, the two languages can be differentiated with very high accuracy of $\sim 99\%$. As a follow-up, Tiedemann and Ljubešić (2012) add Bosnian to the language list showing that most off-the-shelf tools are in no way capable of solving that problem, while their approach by identifying blacklisted words, reaches accuracy of ∼97%. Ljubešić and Klubička (2014) have worked with the same three languages as a subtask of producing web corpora of those languages. They have shown to outperform the best performing classifier from (Tiedemann and

Ljubešić, 2012) by training unigram language models on the whole content of the collected web corpora showing to decrease the error on the Croatian–Serbian language pair to a fourth. Recently, as part of the DSL (Discriminating between Similar Languages) 2014 shared task on discriminating between six groups of similar languages on the sentence level (Zampieri et al., 2014), the language group A consisted of Bosnian, Croatian and Serbian and the best result in the group yielded 93.6% accuracy, which is not directly comparable to the previously reported results because classification was performed on sentence level, and not on document level as in previous research.

To best of our knowledge, there has been only one attempt at discriminating between languages of that level of similarity, namely Croatian and Serbian, on Twitter data in (Ljubešić et al., 2014) where word unigram language models built from the Croatian and Serbian web corpora were used in the attempt at separating users by those two languages. An analysis of the annotation results showed that there is a substantial Twitter activity of speakers of both Bosnian and Montenegrin and that the the collected data cannot be described with the two-language classification schema, but with a 4-class schema which takes into account all the languages in the collection.

Our work builds on top of this previous research by defining a four-language classification schema, inside which Montenegrin, a language that gained official status in 2007, is present for the first time. Additionally, this is the first focused attempt on discriminating between those languages – and possibly between such similar languages overall – on Twitter data.

3. Dataset

The dataset we run our experiments on consists of tweets of 500 random users from the Twitter collection obtained with the TweetCat tool described in (Ljubešić et al., 2014).

There was only one annotator available for this annotation task. Annotating a portion of the dataset by multiple users is considered future work.

Having other languages in the dataset (mostly English) was tolerated as long as most of the text was written in the chosen language. Beside the four main categories, one user, tweeting in Bosnian, had most of the tweets in English (preprocessing error), there was one user tweeting in Macedonian and 8 users were tweeting in Serbian, but using the Cyrillic script. Those 10 users were discarded from the dataset and the following experiments. The users tweeting in Serbian and using the Cyrillic scripts were discarded because we want to concentrate here on discriminating between the languages based on content and not the script used.

The result of the annotation procedure is summarized in the distribution of users given their language presented in Table 1. We can observe that Serbian makes up 77% of the dataset, that there is a similar amount, around 9%, of Bosnian and Croatian data, while Montenegrin is least represented with around 5% of the data. These results are somewhat surprising because there is a much higher number of speakers of Croatian (around 5 million) than of Bosnian (around 2 million) or Montenegrin (below 1 million).

language (code)	# of users
Bosnian (bs)	46
Croatian (hr)	42
Montenegrin (me)	24
Serbian (sr)	378

Table 1: Distribution of users by the language they tweet in

	token	3-gram	6-gram
GNB	0.788	0.769	0.780
KNN	0.780	0.771	0.786
DT	0.894	0.892	0.871
SVM	0.881	0.887	0.897
RF	0.839	0.835	0.843
AB	0.861	0.869	0.876

Table 2: Obtained accuracies in the initial experiments with different classifiers and features

4. Experiments

We perform data preprocessing, feature extraction and data formatting to the svmlight format with simple Python scripts. All the experiments are carried out with the machine learning kit scikit-learn (Pedregosa et al., 2011). Our evaluation metric is accuracy calculated via stratified 5-fold cross-validation.

Each instance in our experiments is one of the 490 annotated Twitter users. We extract features only from the preprocessed text of each user. We could use the information about each specific user like their name, bio, location etc., but we leave this line of research for future work. During preprocessing we remove URLs, hashtags and mentions from the text of each user as well. By preparing our dataset in the described fashion, we remove all the specificities of Twitter generalizing to any sort of user-generated content.

After performing preprocessing, the average number of words per user is 6,606.53 words, with a minimum of 561 and a maximum of 29,246 words.

4.1. Initial experiment

The aim of the initial experiment was to get a feeling for the problem at hand by experimenting with various classifiers and features.

We experiment with the traditional classifiers, such as the Gaussian naive Bayes (GNB), k-nearest neighbor (KNN), decision tree (DT) and linear support-vector machine (SVM), as well as classifier ensembles such as AdaBoost (AB) and random forests (RF). For each classifier we use the default hyperparameter values except for the linear SVM classifier for which we do tune the C parameter for highest accuracy.

From previous research we know that best features for discriminating between similar languages are words and longer character n-grams (around level 6). Traditionally, in the task of language identification, character 3-grams were

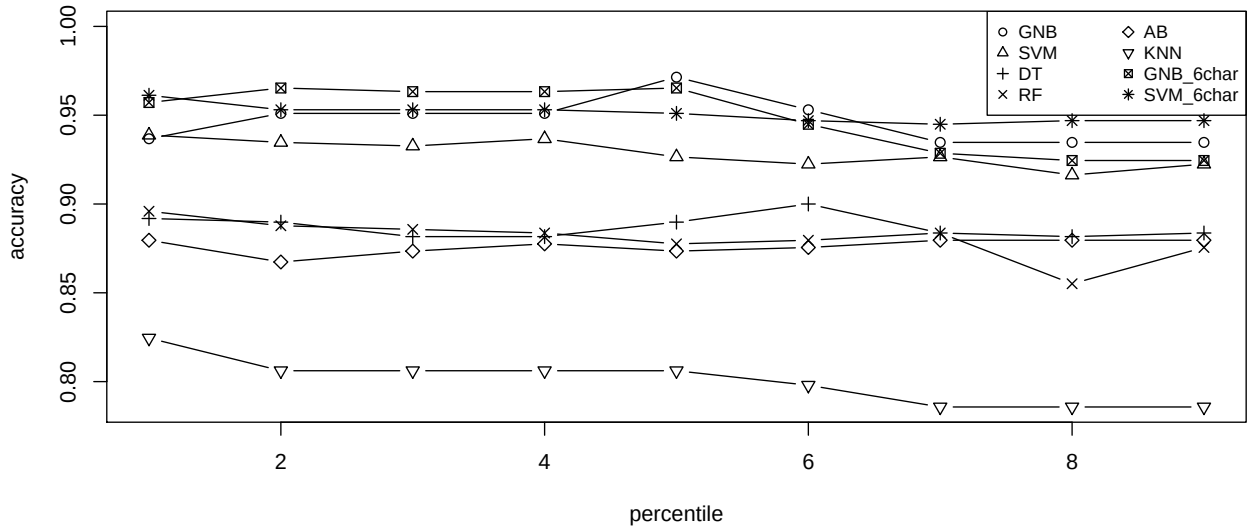

Figure 1: Accuracy of each classifier given the percentile of features with minimal p-values used

most frequently used. This is why we run our initial experiments with three sets of features: tokens, character 3-grams and character 6-grams. We extract character n-grams from tokens with one special character added to the beginning and end of the token. While extracting 6-grams, we add tokens shorter than 4 characters (6 characters with the surrounding special characters) to the feature set as well.

We compare the classifiers by calculating accuracy on 5-fold stratified cross-validation. The results are given in Table 2. We can observe that each set of features produces very similar results, the character 3-gram underperforming slightly, and that the differences between the results are due to usage of a specific classifier. DT and SVM obtain best results while GNB and KNN perform the worst, just slightly above the most-frequent-class baseline. The low score of the GNB classifier, which has no inherent feature selection, and the overall best results obtained by the simple DT classifier, which has implicit feature selection, hint that our results could improve if we applied explicit feature selection as a pre-processing step. This follows our intuition that similar languages can be discriminated through a limited number of features and not the whole lexicon or character n-gram set.

We continue our experiments by introducing a feature selection algorithm and using tokens as our 213,246 initial feature list because of their easier interpretability.

4.2. Feature selection

Although there are stronger feature selection algorithms, we opt for the simple univariate feature selection algorithm which sorts features by their p-value through the F1 ANOVA statistical test and chooses the user-specified percentile of features from the bottom of the list. We use this simple feature selection method because we assume independence of our features, i.e. tokens or character n-grams, which mostly stands in the problem of language identification. Here we experiment with all the classifiers from the previous subsection and the percentile of strongest features ranging from 1 to 9 since all classifiers reach their best performance in that range. The results are shown in Figure 1.

The two best-performing classifiers, once the number of features is down to single-digit percentiles, are the GNB and the SVM. The overall best performing setting is the GNB, which uses 5 percentiles of features (0.971). The worst performing classifier is the KNN which yields worse performance as the number of features increases.

We did perform experiments with other feature sets as well, obtaining very similar results when using character 6-grams (GNB peaking at 2 percentiles with 0.965 and SVM peaking at 1 percentile with 0.961, both depicted in Figure 1) and obtaining worse results when using character 3-grams (0.816 with GNB on 13 percentiles of features and 0.945 with SVM on 4 percentiles of features). Combining the character 6-gram and token feature sets did not produce any improvements, which is to be expected because those two feature sets contain very similar information.

We can consistently observe the phenomenon that SVM outperforms GNB on smaller number of features and on features of lower quality. Although these properties can be important if speed and memory consumption are of great importance, or if no better features are at our disposal, here we choose the GNB on 5 percentiles of features as our final classifier because of its exceedingly superior accuracy. By using 5 percentiles of features, we shrink our model from the initial 213,246 features down to 10,662.

4.3. Confusion matrix and strongest features

We take a closer look at our best performing classifier by plotting our confusion matrix and by calculating preci-

sion and recall on each class. The plot is given in Table 4. We can observe that the two most problematic languages are Bosnian being confused with Serbian and Croatian, and Montenegrin being confused with Serbian.

Next, we inspect the most informative 50 features from our feature selection algorithm and present them, along with the a-posteriori parameter values for each language, in Table 3. While there are a few features that are concept-oriented and not language-specific, such as the toponyms Zagreb and Podgorica (the capitals of Croatia and Montenegro), most features are language-specific and of possible interest to linguists. This is why we will publish all the selected features with the corresponding parameter values for all four languages.

4.4. Learning curve

Finally, we compare our two best-performing classifiers, GNB and SVM by plotting learning curves, using the best performing percentile of features for each classifier. The learning curves are depicted in Figure 2 showing that GNB does outperform SVM on all training data sizes and that there is still room for improvement by moderately increasing the amount of available data.

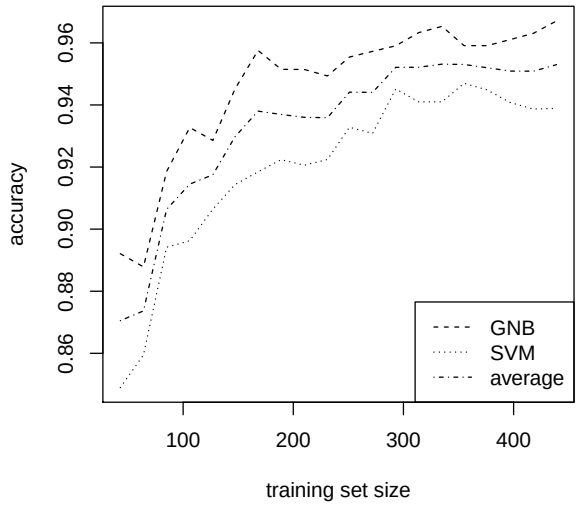

Figure 2: Learning curves of the GNB and SVM classifiers after feature selection

5. Error analysis

We performed error analysis by reinspecting tweets of users that were differently classified by the best performing automated classifier and the human annotator.

We identified altogether 5 users that were incorrectly classified by the human annotator because the bulk of their tweets consisted of retweets and tweets written in languages such as English and German. In those cases, original tweets in the users' native language were very scarce, which made the manual annotation very tiresome. The fact that a third of assumably wrongly classified users are actually human

feature	bs	hr	me	sr
sjutra	0.065	0.048	4.708	0.013
prije	3.152	4.548	5.042	0.119
vrijeme	3.543	5.214	5.292	0.146
dje	1.022	0.0	5.083	0.143
mjesta	0.435	1.19	0.667	0.011
podgorice	0.043	0.0	0.833	0.013
uvijek	4.826	5.69	4.125	0.164
točno	0.022	0.667	0.0	0.003
dio	0.609	1.905	0.625	0.032
cijeli	1.13	1.167	1.292	0.016
poslije	1.87	0.69	1.792	0.048
netko	0.022	2.762	0.0	0.034
gdje	4.0	3.786	1.792	0.101
pg	0.13	0.0	1.875	0.013
tko	0.152	5.357	0.167	0.053
sretan	1.0	2.595	0.042	0.071
podgorica	0.022	0.0	2.0	0.029
cus	0.0	0.0	0.625	0.0
mjesto	0.522	1.119	1.0	0.029
mjestu	0.696	0.571	0.5	0.011
mjeseca	0.391	0.762	0.875	0.008
vjerujem	1.239	0.548	1.042	0.034
lijepo	1.652	2.19	1.5	0.053
zagrebu	0.109	1.643	0.042	0.09
dvije	1.239	1.357	2.5	0.058
ovdje	2.043	1.619	1.0	0.026
podgorici	0.0	0.095	1.0	0.034
vjerovatno	0.5	0.071	0.708	0.005
mjesec	0.891	1.119	1.417	0.045
tjedan	0.0	2.238	0.0	0.003
kuna	0.0	0.881	0.0	0.003
podgoricu	0.0	0.024	0.5	0.008
lijep	0.674	0.595	0.667	0.019
dako	0.022	0.0	0.333	0.0
kava	0.043	1.0	0.042	0.011
bit	0.848	3.857	2.167	0.198
vjerojatno	0.022	0.69	0.0	0.0
ljeto	0.696	1.048	1.75	0.045
pjesme	1.0	0.762	1.333	0.063
umjesto	1.152	0.905	1.167	0.053
kruh	0.0	0.238	0.0	0.0
cg	0.152	0.0	2.625	0.146
zagreb	0.109	2.31	0.0	0.037
svatko	0.0	0.524	0.0	0.011
vidjeti	0.435	0.857	0.292	0.024
negdje	1.13	0.762	0.708	0.019
vazda	0.326	0.0	1.708	0.071
zabolje	0.0	0.0	0.333	0.0
vidji	0.0	0.0	0.375	0.005
pjesma	0.957	0.714	1.25	0.04
djevojkama	0.109	0.024	0.458	0.003

Table 3: 50 strongest features by the univariate feature selection algorithm with per-language parameter values from the GNB classifier

	bs	hr	me	sr	P	R
bs	44	0	0	2	0.917	0.957
hr	1	41	0	0	0.976	0.976
me	0	0	21	3	0.850	0.875
sr	3	1	4	370	0.987	0.979

Table 4: Confusion matrix along with precision and recall for the best performing classifier

annotator errors has motivated us further in including additional annotators in the future.

The remaining 9 manually correctly annotated users were partially wrongly classified because of retweeting. The register in which the users tweet also affected the classification at times. For example, in the almost exclusively colloquial and informal Montenegrin part of the dataset, the only user (a news agency) who tweeted in a more formal register was wrongly classified as belonging to the more inclusive Serbian part of the dataset. It has also been noticed that some users use several languages from the classification schema throughout their tweets, in form of citations and song lyrics. Mixing of these four languages is possible in many contexts, so a dose of indecisiveness in their classification should not be surprising. For that reason we will label each user in our collection not only by the most probable language, but with the distribution of probabilities for all four languages.

6. Conclusion and future work

We have presented a supervised approach to discriminating between very similar languages on Twitter data by classifying each user to the language he or she uses predominantly.

We have annotated 500 users by their predominant language and used that data for experimenting via cross-validation. By using textual features only, we have shown that very similar performance is obtained when using character n-grams or tokens as features. We have shown that feature selection significantly improves the results, which is to be expected given the problem at hand. We obtained very similar results when using linear SVM or Gaussian NB, linear SVM performing better on smaller sets of features or less informative features like character 3-grams, but overall best performance of 97.1% accuracy was obtained using 5% of features and Gaussian NB.

The worst performing language was Montenegrin, being quite often mixed with Serbian, and the second worst Bosnian, being mixed with both Serbian and Croatian.

Next steps include annotating the sample by multiple users for obtaining inter-annotator agreement rates and improving accuracy, as the learning curves suggest. Additionally, at this point only the text of the tweets was used and usage of additional features such as geo-location and user profile information should be inspected as well.

We release the annotated Twitter user lists as well as the prepared datasets in the svmlight format[1] under the CC-BY-SA 4.0 license[2].

7. References

Chu-Ren Huang and Lung-Hao Lee. 2008. Contrastive approach towards text source classification based on top-bag-of-word similarity. In *PACLIC*, pages 404–410. De La Salle University (DLSU), Manila, Philippines.

Nikola Ljubešić and Filip Klubička. 2014. {bs,hr,sr}WaC – web corpora of Bosnian, Croatian and Serbian. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, pages 29–35, Gothenburg, Sweden. Association for Computational Linguistics.

Nikola Ljubešić, Nives Mikelić, and Damir Boras. 2007. Language identification: How to distinguish similar languages. In Vesna Lužar-Stifter and Vesna Hljuz Dobrić, editors, *Proceedings of the 29th International Conference on Information Technology Interfaces*, pages 541–546, Zagreb. SRCE University Computing Centre.

Nikola Ljubešić, Darja Fišer, and Tomaž Erjavec. 2014. TweetCaT: a Tool for Building Twitter Corpora of Smaller Languages. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Lluís Padró and Muntsa Padró. 2004. Comparing methods for language identification. *Procesamiento del Lenguaje Natural*, 33:155–162, September.

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 E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Bali Ranaivo-Malancon. 2006. Automatic Identification of Close Languages – Case study: Malay and Indonesian. *ECTI Transactions on Computer and Information Technology*, 2(2):126–134.

Jörg Tiedemann and Nikola Ljubešić. 2012. Efficient discrimination between closely related languages. In *Proceedings of COLING 2012*, pages 2619–2634, Mumbai, India.

Marcos Zampieri and Binyam Gebrekidan Gebre. 2012. Automatic identification of language varieties: The case of Portuguese. In *Proceedings of KONVENS2012 - The 11th Conference on Natural Language Processing*.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A Report on the DSL Shared Task 2014. In *Proceedings of the VARDIAL workshop*.

[1] http://nlp.ffzg.hr/data/publications/nljubesi/ljubesic14-discriminating/

[2] https://creativecommons.org/licenses/by-sa/4.0/