Similar Language Detection

Shenglan Qiao
Stanford University

Daniel Lévy
Stanford University

Abstract

Language identification is a key component of multiple NLP applications. Discriminating between similar languages is one of the bottlenecks of state-of-the-art language identification systems. In this paper, we present a hierarchical method that first classifies a written sentence into a linguistically defined language group and then determines the language. We are able to achieve an overall test accuracy of 92.6% (12039/13000). Our method is robust and easily scalable to incorporate many languages as well as more training data.

1 Introduction

Language identification has many applications across different domains (translation, language interpretation, etc.). While it is trivial to discriminate between disparate languages, it is far from a solved one for highly similar languages and dialects. Our problem is presented in the Discriminating Similar Language (DSL) Task, a contest that requires teams to predict the language of sentences written in similar languages. The categories can be different languages or variations of the same languages; distinguishing one from another is far from trivial. For example, in the training dataset provided by DSL, more than 20% of the sentences in the South-West Slavic group were written with words present in all three languages, making it impossible to solve with a simple dictionary look-up.

In this paper, we present a hierarchical method that breaks down the problem into two consecutive stages. Languages are grouped by their degrees of similarity into groups. A simple word-frequency method first identifies the group a written text belongs to. A more sophisticated method using an ensemble of Support Vector Machine (SVM) classifiers then determines the language within the group.We have also developed this method with efficiency and scalability in mind. Our computing resources were limited to our personal computers; we had to find a way to keep the size of the data and the complexity manageable without sacrificing too much accuracy.

2 Related Work

Classifying tests written in disparate languages is considered a solved problem. [**?**] presented the word frequency method for language classification, which we used for discriminating among language groups.

Discriminating among similar languages is a fairly present subject in literature and still open for discussion. In September 2015, the LT4VarDial, a joint workshop on Language Technology for closely related language, varieties and dialects was held in Bulgaria. This workshop featured the DSL Task, a contest where teams were given a training set to build models that would be tested on an undisclosed testing set. This shared task inspired several research papers on the subject (an overview can be seen in [**?**]).

[**?**] introduced us to the idea of an ensemble classifier. However, [**?**] included all languages in a single ensemble classifier. We decided this method would be difficult to scale up to include more languages to multiple languages; it would also be computationally inefficient or impossible for us given our limited resources. [**?**] convinced us that a hierarchical methods could perform almost as well as the winner of the task while being more robust and scalable.

3 Dataset and features

The data sets we used were part of the DSL-Shared Task of 2015. The task provided participants with training (17000 examples per language), development (2000 examples per language) and testing sets (1000 examples per languages). The sets consisted of individually labeled sentences extracted from the journalistic corpora in 13 different languages. The languages were part of 6 language groups.

- South-Eastern Slavic (ses): Bulgarian (bg), Macedonian (mk)
- South-Western Slavic (sws): Bosnian (bs), Croatian (hr), Serbian (sr)
- West Slavic (ws): Czech (cz), Slovak (sk)
- Spanish(es): Argentine Spanish (esar), Peninsular Spanish (eses)
- Portuguese (pt): Brazilian Portuguese (ptbr), European Portuguese (ptpt)
- Austronesian (aus): Indonesian (id), Malay (my)

We engineered two categories of features (word and character) and we represented them in a sparse way: **word n-grams** (for example, given a sentence $w_1 \dots w_n$, the 2-grams are $w_1w_2, w_2w_3, \dots, w_{n-1}w_n$) and **character n-grams**: the same than above but with characters instead of words (with character cross-over between words). For example, the 5-grams of ”Hello World” are ”Hello”,”ello ”,”llo W”, ”lo Wo”, ”o Wor”, ” Worl”, ”World”.

We used words n-grams for $n \in \{1, 2\}$ and character n-grams for $n \in \{3, \dots, 6\}$.

4 Methods

The hierarchical method employs a simple word-frequency method that first identifies the group a test belongs to. It then uses an ensemble of SVM models to determine the final output. Figure **??** shows a schematic representation of the method. The SVM models are trained with examples within the language group and can have different number and combination of features across groups.

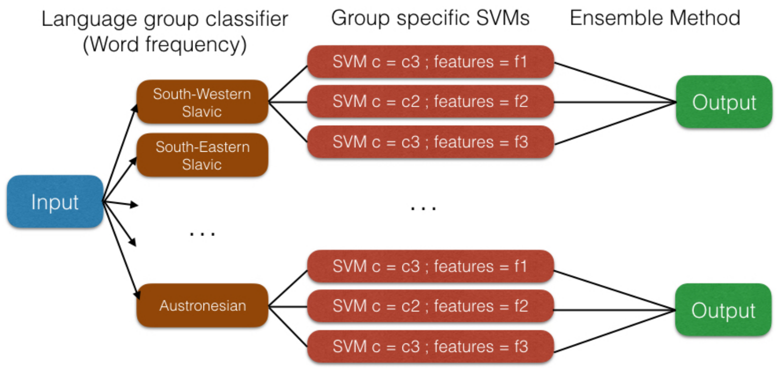

Language group classifier
(Word frequency)
Group specific SVMs
Ensemble Method
Input
South-Western Slavic
South-Eastern Slavic
...
Austronesian
SVM c = c3 ; features = f1
SVM c = c2 ; features = f2
SVM c = c3 ; features = f3
...
SVM c = c3 ; features = f1
SVM c = c2 ; features = f2
SVM c = c3 ; features = f3
Output
Output

Figure 1: Overview of our learning algorithm.

In this section, we will describe the two stages of our hierarchical method as well as our protocol to avoid over-fitting. Parameters of our method were tested and tuned with the development dataset, before applying it to the test dataset.

4.1 The Word-Frequency method

The first block of our pipeline is a classifier to distinguish the language group of an input. We built this classifier using the following word-frequency method. From the training set, for each language

$l \in 1 \dots 13$, we define $x^l \in \mathbb{R}^{1000}$ s.t. x_i^l is the frequency of the ith common word of the lth language.

Given an unclassified sentence s, we define $x^l(s), l = 1 \dots 13$ s.t. $x_i^l(s) = 1$ if the ith word of the lth language is present in the sentence. The classifier is then $h(s) := \underset{l \in 1 \dots 13}{\text{argmax}} \langle x^l(s), x^l \rangle$. To obtain the group, we then just have to return the group the language $h(s)$ is part of.

McNamee showed in [**?**] that this word frequency method was extremely effective when it came to classifying rather distinct languages. This method also proved computationally efficient and obtained virtually perfect result ($> 99\%$ accuracy) for differentiating among language groups, which we will go more in-depth into in section **??**.

4.2 Multi-class SVMs

In class, we derived the optimization program for 2-class SVM. In [**?**], Crammer and Singer makes a derivation for a multi-class SVM which we are using.

Let's replace ourselves in the context of the course but with k classes this time.

Let $(x^{(i)}, y^{(i)}), i = 1 \dots m$ with $x^{(i)} \in \mathbb{R}^n$ but $y^{(i)} \in \{1 \dots k\}$. Crammer and Singer (2002) proposed the following multi-class approach by solving the following optimization problem:

$$\begin{array}{ll} \underset{w_l, \xi_i, l=1\dots k, i=1\dots m}{\text{minimize}} & \frac{1}{2} \sum_{l=1}^{k} w_l^T w_l + C \sum_{i=1}^{m} \xi_i \\ \text{subject to} & w_{y_{(i)}}^T x^{(i)} - w_l^T x^{(i)} \le \delta_{i,l} - \xi_i, \; i = 1, \dots, m. \end{array}$$

Where $\delta_{i,l} = 1$ if $i = l$, $\delta_{i,l} = 0$ if $i \neq l$.

With the following decision function:

$$h_{w_1, \dots, w_k}(x) = \underset{l \in \{1, \dots, k\}}{\text{argmax}} \, w_l^T x$$

We can derive the dual problem:

$$\begin{array}{ll} \underset{\alpha_i^l, l=1\dots k, i=1\dots m}{\text{minimize}} & \sum_{1 \le i,j \le m, 1 \le l \le k} \alpha_i^l \alpha_j^l \langle x^{(i)}, x^{(j)} \rangle + \sum_{1 \le i \le m, 1 \le l \le k} \delta_{i,l} \alpha_i^l \\ \text{subject to} & \sum_{l=1}^{k} \alpha_i^l, \; i = 1, \dots, m. \\ & \alpha_i^l \le \delta_{i,l} C, \forall i \in \{1, \dots, m\}, \forall l \in \{1, \dots k\} \end{array}$$

Where $w_l = \sum_{i=1}^{m} \alpha_i^l x^{(i)}, l = 1 \dots k$

We can then solve the dual using the coordinate ascent method seen in class. It is clear that this extension from 2 classes to k classes works in the exact same way than before and that this classifier has the same properties (kernel trick...). We used LIBLINEAR SVM to implement this multi-class SVM.

4.3 Feature scoring using tf-idf

Character n-grams for higher values ($n \ge 4$ mostly) are very prone to over-fitting. In order to make our method robust, we had to select a subset of features for the ensemble of SVM models. Due to the size of our features set (approximately 3 millions) both PCA and forward/backward search were out of the question. To overcome this obstacle, we decided to score the features using tf-idf and use only the top-rankings ones.

We defined the tf-idf score of a feature t derived from training set D to be

$$\mathrm{tfidf}(t, D) = \mathrm{tf}(t, D) \times \log \frac{N}{\mathrm{df}(t, D)} \tag{1}$$

where $\mathrm{tf}(t, D)$ is the total number of times t appeared in the training set, $\mathrm{df}(t, D)$ the number of examples that contain t, and N the number of training examples. Since $N/df(t, D)$ if always greater or equal to 1, this definition of tf-idf is always non-negative. It is zero if the feature appears in every training example; intuitively, this means the feature is common to all languages in the training set and therefore cannot help distinguish them from one another.

4.4 Ensemble methods

To reduce variance, we also decided to train several SVM models on a given language group and combine them using ensemble methods. For each language groups, we trained p SVM using different hyper-parameters (features and C constant). Each one of these SVM i, outputs k weight vectors. We have $(w_{i,l})_{1 \leq i \leq p, 1 \leq l \leq k}$ weight vectors at our disposition.

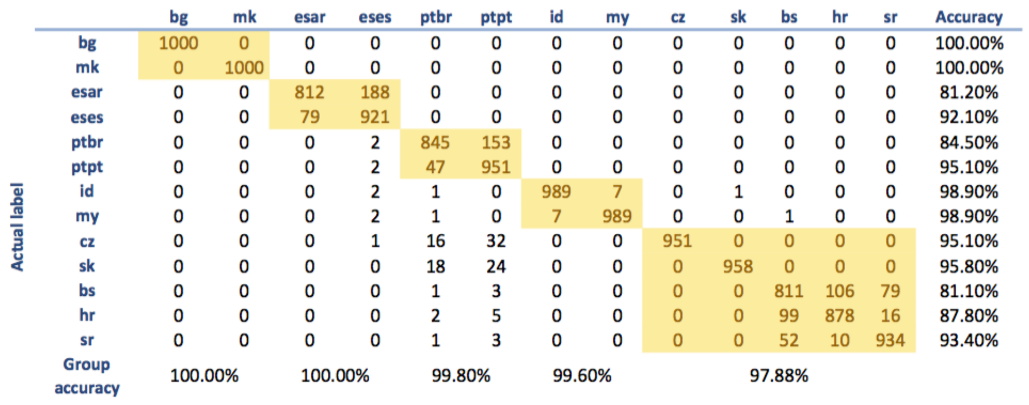
After experimenting with several ensemble methods (majority vote, boosting, and mean confidence) we settled on mean confidence, which can be written as follows:

$$h_{(w_{i,l})_{1 \leq i \leq p, 1 \leq l \leq k}}(x) = \underset{1 \leq l \leq k}{\arg\max} \langle \sum_{t=1}^{p} w_{t,l}, x \rangle$$

For some language group, we did not need to use the ensemble method as high-accuracy classification was straightforward. For the others, we trained and combined 5 SVMs, each had a combination word 1-grams, 2-grams, and character n-grams ($n \in \{3, 4, 5, 6\}$).

5 Results and Discussions

Our final classification results for the test set are shown in figure **??**. Our hierarchical method yields an overall accuracy of 92.6% (12039/13000). For 99.5% of the test examples, the word-frequency method assigns them correctly to their respective language group. We chose to combine the South-Western and West Slavic languages into one group (the sws-ws group) since the word-frequency method tended to group these five languages together.



Actual label	bg	mk	esar	eses	ptbr	ptpt	id	my	cz	sk	bs	hr	sr	Accuracy
bg	1000	0	0	0	0	0	0	0	0	0	0	0	0	100.00%
mk	0	1000	0	0	0	0	0	0	0	0	0	0	0	100.00%
esar	0	0	812	188	0	0	0	0	0	0	0	0	0	81.20%
eses	0	0	79	921	0	0	0	0	0	0	0	0	0	92.10%
ptbr	0	0	0	2	845	153	0	0	0	0	0	0	0	84.50%
ptpt	0	0	0	2	47	951	0	0	0	0	0	0	0	95.10%
id	0	0	0	2	1	0	989	7	0	1	0	0	0	98.90%
my	0	0	0	2	1	0	7	989	0	0	1	0	0	98.90%
cz	0	0	0	1	16	32	0	0	951	0	0	0	0	95.10%
sk	0	0	0	0	18	24	0	0	0	958	0	0	0	95.80%
bs	0	0	0	0	1	3	0	0	0	0	811	106	79	81.10%
hr	0	0	0	0	2	5	0	0	0	0	99	878	16	87.80%
sr	0	0	0	0	1	3	0	0	0	0	52	10	934	93.40%
Group accuracy	100.00%		100.00%		99.80%		99.60%				97.88%			

Figure 2: Confusion matrix for all 13 languages. Group accuracy means the rate at which the word-frequency method correctly identified the language group of test examples.

In order to avoid over-fitting, we tuned the regularization parameter c and performed feature selection based on tf-idf scoring. Within each language group, multiple SVM models were built and tested on the development set. Test accuracies of these model as a function of c were plotted. For each SVM model, we chose the value for c at which the testing accuracy gain was marginal compared to the training accuracy gain.

We selected subsets of features for our final SVM models by observing how their prediction accuracies on the development set changed as a function of features. We ranked features by category (word 1-grams, 2-grams and character n-grams, $n \in \{3, 4, 5, 6\}$) using the tf-idf scores defined in the method section. Starting with the highest ranking features, we computed test accuracies on the development set using model trained with increasing number of features. Figure **??** shows an example of plots we used for feature selection in the sws-ws group. Similar to tuning c, our feature selection protocol kept the top-ranking n features in our final SVM or ensembles (in figure **??** n = 30000) by which increasing the number of features stopped gaining any significant test accuracy on the development set.

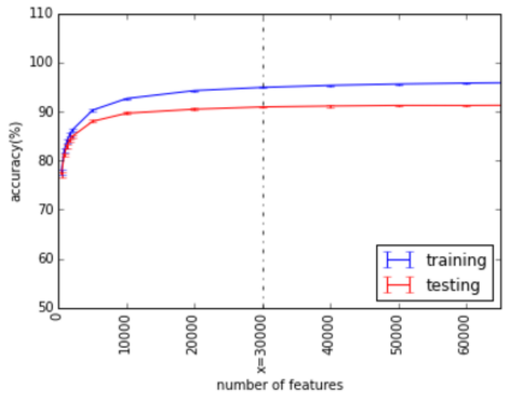


Figure 3: Prediction accuracy on the development set (sws-ws language group) as a function of the number of top-ranking features.

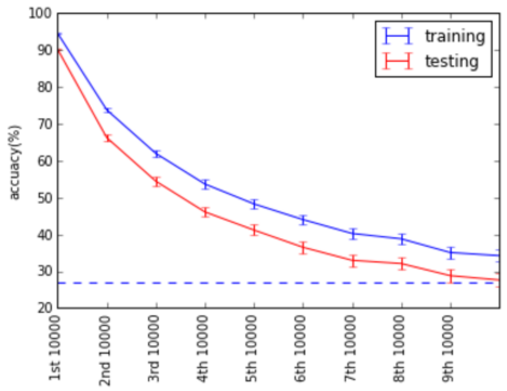


Figure 4: Prediction accuracy on the development (sws-ws language group) set as a function of sets of increasingly lower-ranked SVM features.

Finally, we also demonstrated that the definition of tf-idf was appropriate for feature selection. SVMs built with lower ranking features had lower test accuracies when applied to the development set. For instance, figure **??** shows that test accuracy of models built with sets of word 1-grams decreased monotonically; the first set contained the 10000 top-ranking features, and the successive sets contained the same number of increasingly lower-ranked features. This consistent trend proved that we were avoiding over-fitting by retaining features most representative of the training data.

6 Conclusion

We have achieved classification of 13 similar languages with an overall accuracy of 92.6%. This is close to the DSL Task winning team's accuracy of 95%. More importantly, we have explored a method that is easily scalable and robust. The word-frequency classifier requires maintaining a database of only one thousand frequently occurring words per language. SVM features are derived within a language group, typically made up of only two languages. Generalizing to many languages therefore does not require changing the modeling for existing languages in the hierarchical model but simply adds independent branches to the hierarchy. This advantage of scalability stems from the linguistic knowledge used to divide languages into groups. Additionally, our methodical selection of features and the regularization parameter has rendered our method robustness across the development and test sets. This gives us confidence that given more training data our method will perform well on a variety of texts.

Future work will involve improving performance on certain language groups. Our method prefers to label the test examples as a certain language within the South-Western Slavic, Spanish, and Portuguese groups. More investigation is needed to explain why this happens. Gathering training data from genres other than journalistic articles can also be valuable and may improve the performance of our method.

References

[1] McNamee, P.,2005 *Language identification: a solved problem suitable for undergraduate instruction*. Journal of Computing Sciences in Colleges, 20(3):94–101.

[2] Franco-Salvador, M., Rangel,F.,Rosso,P.,Taule,Marti,M.,A.,2015.*Language variety identification using distributed representations of words and documents*. Proceeding of the 6th International Conference of CLEF on Experimental IR meets Multilinguality, Multimodality, and Interaction (CLEF 2015), volume LNCS(9283). Springer-Verlag.

[3] Zampieri, M., Tan, L., Ljubesic, N., Tiedemann, J., Nakov, P., *Overview of the DSL Shared Task 2015*. Proceeding of the LT4VarDial Workshop.

[4] Malmasi, S., Dras, M., *Language Identification using Classifier Ensembles*. Proceeding of the LT4VarDial Workshop.

[5] Franco-Salvador, M., Rosso, P., & Rangel, F. 2015., *Distributed Representations of Words and Documents for Discriminating Similar Languages*. Proceeding of the LT4VarDial Workshop

[6] Acs, J., Grad-Gyenge L., Bruno Rodrigues de Rezende Oliveira, T. 2015. *A two-level classifier for discriminating similar languages*. Proceeding of the LT4VarDial Workshop

[7] Crammer, K., Singer, Y. 2002 *On the Learnability and Design of Output Codes for Multiclass Problems*. Machine Learning, Vol. 47.