Discriminating between Similar Languages Using PPM

Victoria Bobicev
Technical University of Moldova
victoria_bobicev@rol.md

Abstract

The paper presents the results of participation of Bobicev team in DSL (Discriminating Similar Languages) shared task 2015. It describes the use of PPM (Prediction by Partial Matching) for language discrimination. The accuracy of the presented system was equal to 94.14% for the first set and 92.22% for the second set. The results were scored as the 4th for the first task and 5th for the second task, the best results being 95.54% and 94.01% respectively.

1 Introduction

The task of language identification is the problem of detection what language a document is written in. The task seems to be relatively easy and many statistical methods achieve relatively high accuracy (more than 95%) for language detection. However, the good results obtained in the laboratory simplified conditions become worse in the real word circumstances. Very short documents (such as tweets), fragments of various languages in one text, documents written in similar languages – here are just some difficulties encountered by the language detection systems. The present paper describes use of PPM (prediction by Partial Matching) statistical method for language discrimination task.

The accuracy of the presented system for the DSL 2015[1] (Discriminating Similar Languages) shared task (Zampieri et al., 2015) was equal to 94.14% for the first set; 92.22% for the second set respectively. The results were scored as the 4th for the first task and 5th for the second task, the best results being 95.54% and 94.01% respectively.

The advantage of the proposed method is its relative simplicity. The method operates with sequences of characters or even bytes, thus it does not need to tokenize or preprocess the analyzed text in any way. This also makes it relatively fast in training and text processing.

The paper is organized as follows: the next part gives a short overview of the related work; section 3 contains the system description and explanations how it was used for the task at hand; section 4 includes task (4.2) and data presentation (4.1), experiments and the obtained results (4.3, 4.4). Finally, a discussion concludes the paper.

2 Related work

The first DSL (Discriminating Similar Languages) shared task has been organized in 2014 and the task participants presented their systems at the VarDial workshop at COLING 2014. The DSL corpus collection was created for the evaluation by merging three comparable corpora of similar languages and language varieties. Tan et al. (2014) described the process of the corpus creation and reported the performance of up to 87.4% accuracy for the baseline discrimination experiments. In the overall report for this task (Zampieri, 2014) the organizers presented the results of 8 final submissions. All participants that described their systems used statistical methods such as Naïve Bayes, SVM, Max. Ent. and other. All of them used words and character n-grams as features.

The shared task organizers mentioned that the problem of similar languages discrimination was similar to the problem proposed in the Native Language Identification (NLI) shared task (Tetreault et al., 2013) where participants were provided English essays written by foreign students of 11 different mother tongues and had to identify the native language of the writer of each text. The differences between very similar languages can be as subtle as in case of the same language used by different people.

Ljubešić & Kranjčić (2014) presented the work on discrimination between tweets written in very similar languages, namely Bosnian, Croatian,

[1] http://ttg.uni-saarland.de/lt4vardial2015/dsl.html

Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects, pages 59–65,
Hissar, Bulgaria, September 10, 2015.

Montenegrin and Serbian and testing a number of statistical methods and various features such as tokens, character 3-grams and 6-grams obtained the best accuracy of ~97%. The authors mentioned that in some cases the text can be written in a mixture of languages either similar ones or with fragments of English or other widely used languages.

Baldwin & Lui (2010) analyzed the influence of number of discriminated languages, the amount of training data and the length of documents on the accuracy of document language detection. They experimented with three relatively difficult corpora: (1) EUROGOV containing relatively longer documents, all in a single encoding, spread evenly across a relatively small number (10) of Western European languages; (2) TCL (Thai Computational Linguistics Laboratory) with a larger number of languages (60) across a wider range of language families, with shorter documents and a range of character encodings; (3) WIKIPEDIA: a slightly larger number of languages (67), a single encoding, and shorter documents. Testing a number of statistical methods and using bytes, codepoints (pairs of bytes), uni-, bi-, and trigrams as features they obtained the best accuracy 0.987 for EuroGOV; 0.977 for TCL and 0.671 for Wikipedia. Experimenting with the n-grams of various length they managed to rise the accuracy to 0.729 for Wikipedia. The authors found that longer documents were easier for detection however they often contained fragments in other languages different than the main language of the document.

Malmasi (2015) presented the work on discriminating two similar languages: Persian and Dari achieving the 96% accuracy using character and word n-grams on the collected corpus of 28k sentences (14k per-language). Out-of-domain cross-corpus evaluation, however, achieved 87% accuracy in classifying 79k sentences from the Uppsala Persian Corpus.

3 System description

We explored the PPM (Prediction by Partial Matching) model for automatic text language detection. Prediction by partial matching (PPM) is an adaptive finite-context method for text compression that is a back-off smoothing technique for finite-order Markov models (Bratko et al., 2006). It obtains all information from the original data, without feature engineering, it is easy to implement and relatively fast. PPM produces a language model and can be used in a probabilistic text classifier. Treating a text as a string of characters, the character-based PPM avoids defining word boundaries; it deals with different types of documents in a uniform way. It can work with texts in any language and be applied to diverse types of classification.

PPM is based on conditional probabilities of the upcoming symbol given several previous symbols. A blending strategy for combining context predictions is to assign a weight to each context model, and then calculate the weighted sum of the probabilities:

$$P(x) = \sum_{i=1}^{m} \lambda_i p_i(x), \qquad (1)$$

where λ_i and p_i are weights and probabilities assigned to each order *i (i=1…m)*.

For example, the probability of character '***m***' in context of the word '***algorithm***' is calculated as a sum of conditional probabilities dependent on different context lengths up to the limited maximal length:

$$P_{PPM}(\text{'}\boldsymbol{m}\text{'}) = \lambda_5 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{orith}\text{'}) + \lambda_4 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{rith}\text{'}) + \\ + \lambda_3 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{ith}\text{'}) + \lambda_2 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{th}\text{'}) + \\ + \lambda_1 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{h}\text{'}) + + \lambda_0 \cdot P(\text{'}\boldsymbol{m}\text{'}) + \lambda_{-1} \cdot P(\text{'esc'}),$$

where

λ_i (i = 1…5) is the normalization weight;

5 is the maximal length of the context;

P('esc') is so called 'escape' probability, the probability of an unknown character.

PPM is a special case of the general blending strategy. The PPM models use an escape mechanism to combine the predictions of all contexts of all lengths starting with the maximal length m and ending with the context -1.

The PPM escape mechanism is more practical to implement than weighted blending. In the general weighted blending the weighted coefficients have to be estimated and this requires additional calculations. In PPM the escape mechanism replaces the coefficients. The estimation of a character probability starts with the context of the maximal length m. If the given character probability can be estimated with this context, this probability is used for the character. If this context has not appeared and the character probability cannot be estimated with the longest context m, the method moves to the shorter context m-1 using the escape mechanism. If the shorter context also cannot be used, the method moves to the shorter context. Context -1 ensure that this happens even in the case when the character itself is unknown in the model.

There are several versions of the PPM algorithm depending on the way the escape probability for each context is estimated. In our implementation, we used the escape method C, named PPMC; more details can be found in (Bobicev, 2007). The maximal length of a context equal to 5 in PPM model was proven to be optimal for text compression (Teahan, 1998). In all our experiments with character-based PPM model we used maximal length of a context equal to 5; thus our method is PPMC5.

As a compression algorithm PPM is based on the notion of *entropy* introduced as a measure of a message uncertainty (Shannon, 1948):

$$H_d = -\sum_{i=1}^{n} p(x_i)\log p(x_i) \quad (2)$$

where

H_d – entropy of text *d*;

$p(x_i)$ - probability of character x_i ($i = 1 \ldots n$) for all characters in the text *d*.

Cross-entropy is the entropy calculated for a text if the probabilities of its characters have been estimated on another text (Teahan, 1998):

$$H_d^m = -\sum_{i=1}^{n} p^m(x_i)\log p^m(x_i) \quad (3)$$

where

n is the number of symbols in a text d,

$H_d{}^m$ is the entropy of the text d obtained by model m,

$p^m(x_i)$ is a probability of a symbol x_i in the text *d* obtained by model m.

The cross-entropy between two texts is greater than the entropy of a text itself, because probabilities of characters in diverse texts are different:

$$H_d^m \geq H_d \quad (4)$$

The cross-entropy can be used as a measure for document similarity; the lower cross-entropy for two texts is, the more similar they are. Hence, if several statistical models had been created using documents that belong to different classes and cross-entropies are calculated for an unknown text on the basis of each model, the lowest value of cross-entropy indicates the class of the unknown text. In this way cross-entropy is used for text classification.

In practical tasks the per-character entropy is used in order to avoid the influence of document length in the process of entropy comparison:

$$H_L = \frac{1}{n}\left(-\sum_{i=1}^{n} p(x_i)\log p(x_i)\right)$$

Our utility function for text classification was per-character cross-entropy of the test document while the probabilities were estimated on the base of the known classes of documents.

On the training step, we created PPMC5 models for each class of documents; on the testing step, we evaluated cross-entropy of previously unseen texts using models for each class. Thus, cross-entropy was used as similarity metrics; the lowest value of cross-entropy indicated the class of the unknown text.

There are several variations of PPM method. One possible is to use not all characters from the text but only some of them, for example, only alphanumeric characters or only letters. In our case when we have to discriminate the languages not all characters in text seem important. We probably do not need any figures or special characters but the punctuation may be the specific for the language.

Another variation is the word-based PPM (Bobicev, 2006). For some tasks words can be more indicative text features than character sequences. That's why we decided to try both character-based and word-based models for language identification. In the case of word-based PPM, the context is only one word and an example for the formula (1) looks like the following:

$$P_{PPM}(\text{'word}_i\text{'}) = \lambda_1 \cdot P(\text{'word}_i\text{'} \mid \text{'word}_{i-1}\text{'}) + \lambda_0 \cdot P(\text{'word}_i\text{'}) + \lambda_{-1} \cdot P(\text{'esc'}),$$

where

$word_i$ is the current word;

$word_{i-1}$ is the previous word.

This model is coded as PPMC1 because of the same C escape method and one length context used for probability estimation.

4 Experiments description

The experiments were carried out during the DSL 2015 shared task event. The first set of the experiments was performed on the base of training data released by the organisers in May 2015. The second set consisted of evaluation runs on test data released in June and the results for these experiments were provided by the organizers.

4.1 The Data Description

For the DSL shared task 2015 edition, the organizers released two new versions of the DSL corpus collection[2] (DSLCC), the version 2.0 and 2.1[3]. The version 2.0 is the standard shared task training material whereas the version 2.1 can be

[2] https://bitbucket.org/alvations/dslsharedtask2014

[3] http://ttg.uni-saarland.de/lt4vardial2015/dsl.html

used for the unshared task track or as additional training material. The collection is described in (Tan et al., 2014).

In 2015, apart from the similar languages and varieties the training and test sets were also including texts from other languages to emulate a real-world language identification scenario. Finally, the two released versions were the following:

1) DSLCC version 2.0. contained Bulgarian, Macedonian, Serbian, Croatian, Bosnian, Czech, Slovak, Argentinian Spanish, Peninsular Spanish, Brazilian Portuguese, European Portuguese, Malay, Indonesian and a group containing texts written in a set of other languages.
2) DSLCC version 2.1. contained all the DSLCC version 2.0. plus Mexican Spanish and Macanese Portuguese.

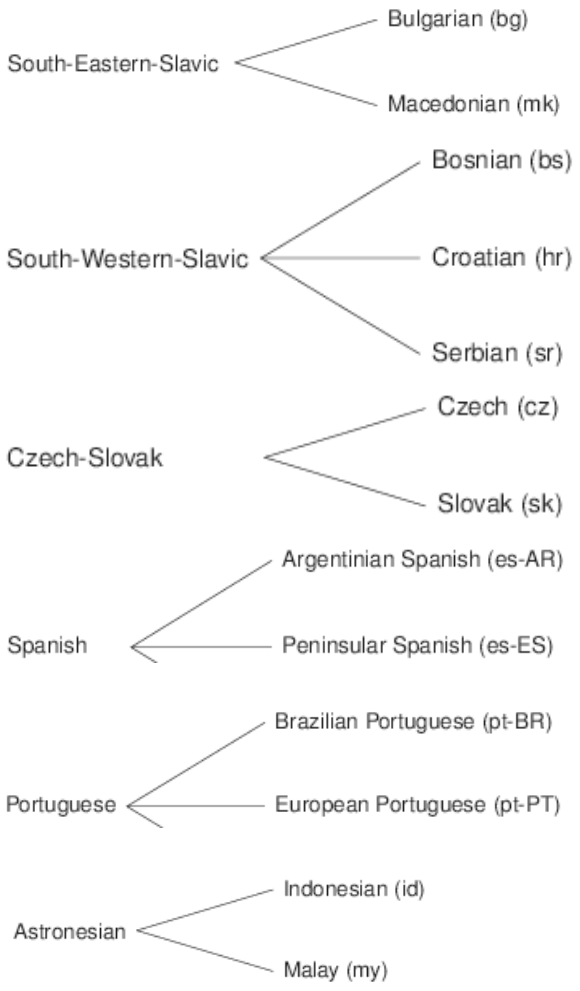

Figure 1. Groups of similar languages which presented difficulties in the process of language identification.

The corpus contained 20,000 instances per language (18,000 training + 2,000 development). Each instance was an excerpt extracted from journalistic texts containing 20 to 100 tokens and tagged with the country of origin of the text. The groups of similar languages are presented in figure 1.

4.2 The task description

For the testing phase two test sets (A and B) have been released. Each of them contained 1,000 unidentified instances of each language to be classified according to the country of origin. These instances contained also instances of texts from the other languages than those presented in the figure similarly to the training set DSLCC version 2.0.

Test set A contained original unmodified newspaper texts. Test set B contained modified newspapers texts processed with NER taggers to substitute named entities for place holders.

Participants had to return their results in up to 2 days after the release of the test sets. Scores were calculated according to the systems' accuracy in identifying the country of origin of the text. Two kinds of submissions were allowed:

1) Closed submission: Using only the training corpus provided by the DSL shared task (DSLCC v.2.0).

2) Open submission: Using any corpus for training including or not the DSLCC v.2.0.

We participated only in closed submission using just the corpus DSLCC v.2.0.

4.3 The first set of the experiments

In order to evaluate the PPM method for the task we used 10-fold cross-validation on the all provided training data. Initially, we excluded the instances marked as xx with the unknown languages to see how the method performed on the known sets. Thus, for each step we used 1800 instances of each language for training and 200 instances of each language for test.

We used character-based PPM5 in the first set of the experiments. The first experiment was performed using only letters for training, all other characters were ignored.

metrics	experiment	
	1st	2nd
microaverage F-score	0.928	0.933
macroaverage Precision	0.929	0.934
macroaverage Recall	0.928	0.933
macroaverage F-score	0.928	0.933

Table 1: The results for the first and the second experiments using letter and character based PPM5

In the second experiment we used all characters from the texts; all letters were converted in lower case. The results for the first and the second experiments are presented in table 1.

Thus, we obtained slightly better results in case when all characters from the texts were used.

The next experiment was performed using word-based PPM1 described in the previous section. Its results were worse than for character based PPM5.

Next, we experimented with the unknown languages marked as xx. There were several languages and thus, they did not present one uniform class of texts. While the entropy for the texts written in the same language was in average 2 ± 0.5 bit/symbol, for the different languages the average entropy varied from 4 to even 12 bit/symbol.

We considered two options of the unknown languages classification:

- A threshold for these languages was used. If the smallest entropy of a test text on the base of all models is bigger than a threshold we considered that text as not written in any of 13 known languages and hence, as unknown one and marked as xx. In this case we created models only on 13 known classes of languages.
- All text marked with xx were treated as one more class. In this case, 14 models were created, including a model for xx class and the standard procedure was applied. Each test document was attributed to the class for which it has the lowest entropy.

metrics	experiment	
	1st	2nd
microaverage F-score	0.922	0.938
macroaverage Precision	0.926	0.939
macroaverage Recall	0.922	0.938
macroaverage F-score	0.924	0.939

Table 2: The results for the first and the second experiments with the unknown texts marked as xx

The obtained results are presented in table 2. Thus, the second option when all the documents written in the unknown languages were treated as the one class was better. More than that, this result was even better than the pure classification of 13 known languages. This indicates that xx class was distinguished fairly well.

4.4 The second set of the experiments

The second set of the experiments was performed on the base of the test data released by the organizers of DSL shared task in June 2015.

These DSL Test Sets are part of the DSLCC v2.0, they comprise news data from various corpora to emulate the diverse news content across different languages and varieties.

Two types of test data were released:

•The first test set that contained 14,000 unchanged sentences for 13 anguages/varieties and others (bg, bs, cz, es-AR, es-ES, hr, id, mk, my, pt-BR, pt-PT, sk, sr, xx).

•The second test that contained 14,000 sentences with that had blinded Named Entities. In these texts, the Named Entities (NEs) have been replaced by placeholders; a #NE# instead of a named entity.

An example of such sentence is:

The initial sentence: La cinta, que hoy se estrena en nuestro país, competirá contra Hors la Loi, de Argelia, Dogtooth, de Grecia, Incendies, de Canadá, Life above all , de Sudáfrica, y con la ganadora del Globo de Oro, In A Better World, de Dinamarca.

The sentence with blinded NE: La cinta, que hoy se estrena en nuestro país, competirá contra #NE# la #NE# , de #NE# , #NE# , de #NE# , #NE# , de #NE# á, #NE# above all , de #NE# , y con la ganadora del #NE# de #NE# , #NE# A #NE# #NE# , de #NE# .

The participants were allowed to submit only 3 runs for closed and/or 3 runs for open task for both test sets.

We submitted only one run for each test set using PPM5 character based method using all characters from the text as this option demonstrated the best results in the first set of experiments. While experimenting with the second test set with blinded NE we simply removed #NE# fragments and worked with the rest of the text. Thus, the example of the sentence presented above would look as follows: La cinta, que hoy se estrena en nuestro país, competirá contra la , de , , de , , de á, above all , de , y con la ganadora del de , A , de .

The overall accuracies in these experiments were calculated by the organizers as such:

overall accuracy = sum(TP) / #sents

where:

TP = True Positive for all languages/varieties;

#sents = total number of documents in evaluation dataset.

The accuracy for the first task was equal to 94.14; for the second set it was 92.22. The results were scored as the 4th for the first task and 5th for

the second task, the best results being 95.54 and 94.01 respectively.

5 Discussion

The challenges are an excellent way to examine the problem at hand from the various points of view. The challenge organizer's work is very important in this context. The saying is that the good question contains a part of the answer. In the case of the challenge the findings depend heavily on the quality of the prepared data. It should be mentioned though that the flaws in the data preparation could lead to interesting discoveries as well.

In this particular challenge the problem was to discriminate between similar languages. The organizers indicated the groups; figure 1 presents them. The best way was to analyze the accuracy on every group apart; this information was not provided for the final test. We present and discuss it on the base of the 10-fold cross-validation experiment that used the 260,000 training instances.

languages	bg	mk
bg	19996	3
mk	1	19997

Table 3: Confusion table for Bulgarian and Macedonian

As it is seen from the table, Bulgarian and Macedonian can be reliably distinguished due to several specific characters in Macedonian alphabet which are frequent enough to appear in any sentence despite of the similarity of these two languages in both in vocabulary and syntax. A couple of misclassified sentences were written in a special manner; here is an example: "При то-зи из-раз ве-че яс-но си про-ли-ча, как да-ма-та уми-ш-ле-но вмъ-к-ва ня-ка-къв ак-цент, с дру-ги ду-ми бъл-гар-с-ки-ят й ве-че та-ка убя-г-ва, та чак го фъ-ф-ли."

languages	bs	hr	sr
bs	16168	2637	1195
hr	2977	16797	226
sr	2118	403	17479

Table 4: Confusion table for Bosnian, Croatian and Serbian

The worst results were obtained for the group of Bosnian, Croatian and Serbian languages as they overlap significantly in vocabulary, syntax and morphology. Although they claimed to be different languages the differences were not so frequent and easily identified as in case of Bulgarian and Macedonian. The most overlapping were Bosnian and Croatian; 13% of Bosnian sentences were classified as Croatian and 15% of Croatian sentences were classified as Bosnian.

languages	es-AR	es-ES
es-AR	17547	2453
es-ES	1607	18391

Table 5: Confusion table for Argentinean Spanish and Peninsular Spanish

The two Spanish dialects discrimination results were better than for Slavic languages; 12% of Argentinean Spanish sentences were classified as Peninsular Spanish and 9% of Peninsular Spanish sentences were classified as Argentinean Spanish. The differences here were also not so frequent and in many sentences were no any specific feature to help the source detection.

languages	pt-BR	pt-PT
pt-BR	18440	1558
pt-PT	1978	18021

Table 6: Confusion table for Brazilian Portuguese and European Portuguese

The situation for Brazilian Portuguese and European Portuguese was similar; 8% of Brazilian Portuguese sentences were classified as European Portuguese and 11% of European Portuguese sentences were classified as Brazilian Portuguese.

languages	id	my
id	19905	93
my	177	19823

Table 7: Confusion table for Indonesian and Malay

The differences between Indonesian and Malay are much more frequent and easily learned by a statistical system; less than 1% of sentences were misclassified.

It should be noted that the instances written in unknown languages and marked as xx were classified almost perfectly. Only several sentences were classified as Spanish but they seemed to be the Spanish ones; for example: "El manifiesto del Consell de la Llengua empieza afirmando que la

lengua catalana "constituye una fuente de igualdad de oportunidades y de cohesión social en Balears".

The discussion raised in the corpora list disputed the question: has the problem of language discrimination finally been solved? The answer is no. Probably, the question should be reformulated as follows: is it even possible to obtain 100% correct discrimination between the languages, especially similar ones? And the answer would be again no. Languages are a part of the constantly changing world, so they also tend to be highly dynamic. Some languages disappear, some appear, some split, and some merge due to linguistic researches or political changes. For example, while we were solving the discrimination task between Serbian and Croatian but many linguists consider Serbian and Croatian to be dialects of one language, not separate languages and refer to it as Serbo-Croatian. The paper by Xia et al., (2010) presented an example of the complexity of language discrimination tasks. They presented a table of language names for which they could not even find a standard language ID code. There were also "missing" and ambiguous language names; tables of 1-to-n split of languages. They pointed out that our knowledge of languages is always changing and expanding, which entails the need of annual revision of the language list.

A good example of all said above is Moldavian language, which has been declared the official language with the new Moldovan Cyrillic alphabet due to political changes (appearance of Moldavian Republic as a part of Soviet Union). The differentiation of Moldavian and Romanian languages was introduced in the context of the Soviet policy that emphasized the differences between Moldova and Romania. Its existence is officially denied now because the current Moldavian government declared Romanian language as the official one in the Republic of Moldova. As in many other cases the new language was not linguistically but purely politically motivated. The linguists don't even want to delve deeper into that matter because there are many conflicting interests - political, cultural and even financial.

The other, pure practical question is: do we really need to obtain 100 percent accuracy in this task? The answer is also no. If the languages are really close some sentences are impossible to detect reliably; they could be written in any of related language and any language tool adapted to one of these languages is able to analyze it satisfactory.

References

Baldwin, T., Lui, M. 2010. *Language identification: The long and the short of the matter* (2010) In Proc. HLT-NAACL.

Bobicev, V. 2006. *Text Classification Using Word-Based PPM Models, The Computer Science Journal of Moldova*, vol. 14, no. 2, pp. 183–201.

Bobicev, V. 2007 *Comparison of Word-based and Letter-based Text Classification*. RANLP V, Bulgaria, pp. 76–80.

Bratko A., Cormack G. V., Filipic B., Lynam T. R., Zupan B. 2006. *Spam filtering using statistical data compression models*, Journal of Machine Learning Research 7:2673–2698.

Ljubešić, N., Kranjčić, D. 2014. *Discriminating between VERY similar languages among Twitter users.* 9th Language Technologies Conference Information Society – IS.

Malmasi, S. 2015. *Discriminating Similar Languages: Persian and Dari.* Volume 3 of Tiny Transactions on Computer Science.

Shannon, C. E. 1948. *A Mathematical Theory of Communication*. The Bell System Technical Journal, vol. 27, pp. 379–423, 623–656.

Tan, L., Zampieri, M., Ljubešic, N., Tiedemann, J. 2014. *Merging Comparable Data Sources for the Discrimination of Similar Languages: The DSL Corpus Collection.* Proceedings of the 7th Workshop on Building and Using Comparable Corpora (BUCC), Iceland.

Teahan, W. 1998. *Modelling English text*, PhD Thesis, University of Waikato, New Zealand.

Tetreault, J., Blanchard, D., Cahill, A. 2013. *A report on the first native language identification shared task.* In Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, Atlanta, GA, USA. Association for Computational Linguistics.

Xia, F., Lewis, C., Lewis, W. D. 2010. *The Problems of Language Identification within Hugely Multilingual Data Sets*, Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010).

Zampieri, M., Tan, L., Ljubešić, N., Tiedemann, J. 2014. *A Report on the DSL Shared Task 2014.* 1st Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects (VarDial), Ireland.