Discriminating between Similar Languages Using a Combination of Typed and Untyped Character N-grams and Words

Helena Gómez-Adorno[1], **Ilia Markov**[1], **Jorge Baptista**[2], **Grigori Sidorov**[1], **David Pinto**[3]

[1]Instituto Politécnico Nacional, Center for Computing Research,
Av. Juan de Dios Bátiz, C.P. 07738, Mexico City, Mexico
[2]Universidade do Algarve/FCHS and INESC-ID Lisboa/L2F,
Campus de Gambelas, P-8005-139, Faro, Portugal
[3]Benemérita Universidad Autónoma de Puebla, Faculty of Computer Science,
Av. San Claudio y 14 Sur, C.P. 72570, Puebla, Mexico
helena.adorno@gmail.com, markovilya@yahoo.com,
jbaptis@ualg.pt, sidorov@cic.ipn.mx, dpinto@cs.buap.mx

Abstract

This paper presents the CIC_UALG's system that took part in the Discriminating between Similar Languages (DSL) shared task, held at the VarDial 2017 Workshop. This year's task aims at identifying 14 languages across 6 language groups using a corpus of excerpts of journalistic texts. Two classification approaches were compared: a single-step (all languages) approach and a two-step (language group and then languages within the group) approach. Features exploited include lexical features (unigrams of words) and character n-grams. Besides traditional (untyped) character n-grams, we introduce typed character n-grams in the DSL task. Experiments were carried out with different feature representation methods (binary and raw term frequency), frequency threshold values, and machine-learning algorithms – Support Vector Machines (SVM) and Multinomial Naive Bayes (MNB). Our best run in the DSL task achieved 91.46% accuracy.

1 Introduction

Discriminating between Similar Languages (DSL) is a Natural Language Processing (NLP) task aiming at automatically identifying the language in which a text is written. From the machine-learning perspective, DSL can be viewed as a multi-class, single-label classification problem, in which automatic methods have to assign class labels (languages) to objects (texts). DSL can be used in a variety of applications, including security and forensics, when, for example, identifying the language/dialect in which a given threat is written can help limit the search space of the author of this threat. Moreover, automated DSL is a useful aid for machine translation and information retrieval systems.

Discriminating between Similar Languages (DSL) shared task[1] provides a common platform for researchers interested in evaluating and comparing their systems' performance on discriminating between similar languages. The DSL 2017 edition (Zampieri et al., 2017) focuses on a set of 14 language varieties within 6 language groups using short text excerpts extracted from journalistic texts. Similar languages or language varieties are grouped by similarity or by their common origin.

According to (Malmasi and Dras, 2015; Çöltekin and Rama, 2016; Jauhiainen et al., 2016; Zirikly et al., 2016), high-order character n-grams and their combinations have proved to be highly discriminative for the DSL task, hence this study examines the variation of n from 1 to 6 on untyped (traditional) n-grams, but foremost this work introduces in this task the use of typed character n-grams (with n varying between 3 and 4), that is, character n-grams classified into the categories introduced by Sapkota *et al.* (2015). The authors defined 10 different character n-gram categories based on affixes, words, and punctuation. Typed character n-grams have shown to be predicative features for other classification tasks, such as Authorship Attribution (Sapkota et al., 2015) and Author Profiling (Maharjan and Solorio, 2015), including a cross-genre scenario (Markov et al., 2016). To the best of our knowledge, this is the

[1]http://ttg.uni-saarland.de/vardial2017/sharedtask2017.html

Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 137–145,
Valencia, Spain, April 3, 2017.

first time typed character n-grams are used in the DSL task.

Furthermore, a single-step and a two-step classification approaches were built. In the single-step approach, all 14 languages are discriminated against each other. In the two-step approach, first, the language group is predicted, and then the language variety within the group. Besides, two different feature representation methods were tested, namely, binary feature representation and term frequency weighting scheme. Several threshold values were evaluated in order to fine-tune the feature set for the final submission. Finally, the performance of two popular machine-learning algorithms was examined: Support Vector Machines (SVM) and Multinomial Naive Bayes (MNB).

The remainder of the paper is organized as follows: Section 2 discusses the related work. Section 3 presents the proposed methodology. First, subsection 3.1 provides some characteristics of the DSL 2017 corpus, and subsection 3.2 describes the conducted experiments. Section 4 provides the obtained results and their evaluation. Next, Section 5 discusses these results in the light of the typed n-gram features, newly introduced in the DSL task, and based on the results from the experiments carried out on the development set. Section 6 draws the conclusions and points to possible directions of future work.

2 Related Work

The task of identifying the language of a text has been largely studied, and it is considered a solved problem. However, recent studies have shown that the task is more difficult when the texts are from different domains and have different lengths (Lui and Baldwin, 2011), when they contain code-switching (Solorio et al., 2014), or when the texts are very similar (Tan et al., 2014).

Motivated by the shared task on Discriminating between Similar Languages (DSL), there has been an increasing number of published papers in this research field. The organizers of the task compiled and released the DSL *Corpus Collection* (DSLCC) (Tan et al., 2014), which includes short excerpts from journalistic texts. It is divided, according to the version, in groups of languages. The different versions of the corpus can be found in the corresponding overview papers of the DSL task (Zampieri et al., 2014; Zampieri et al., 2015; Malmasi et al., 2016). The DSL shared task offers closed and open tracks; the open track allows the use of additional information or material apart from the provided training corpus, whereas the closed track only allows the use of the provided training corpus. The rest of the section will focus on the related work on the closed DSL tasks.

Most of the work on the DSL research topic addresses the task as a classification problem, using supervised machine-learning algorithms. The best performing methods for DSL use high-order character *n*-gram and word *n*-gram features (Goutte et al., 2016; Ionescu and Popescu, 2016). For a complete guide of the approaches developed for the DSL shared task, please refer to the overview papers of each edition (Zampieri et al., 2014; Zampieri et al., 2015; Malmasi et al., 2016).

In the fist edition of the DSL shared task (Zampieri et al., 2014), the best performance was achieved by the NRC-CNRC (Goutte et al., 2014) team. They proposed a two-step classification approach to predict first the language group and then the languages within the group. For both steps, they used Support Vector Machines (SVM) with word and character n-gram features. In the 2015 edition of the DSL shared task, the best performing system (Malmasi and Dras, 2015) proposed an ensemble of SVM classifiers, each trained on a single feature type. The used feature types include character n-grams (n = 1–6), word unigrams, and word bigrams. In the 2016 edition of the task, the winning approach (Çöltekin and Rama, 2016) used a single SVM classifier with linear kernel trained on character *n*-gram features of length from 1 to 7. The winning team also reported additional experiments with deep learning architectures, concluding that the linear models perform better in the DSL task.

In summary, DSL approaches can be divided into single- and two-step classification; the most popular machine-learning algorithms for this task are SVM, Logistic Regression, and ensemble classifiers. Other techniques have been also explored in the DSL task, including token-based backoff (Jauhiainen et al., 2016), prediction by partial matching (Bobicev, 2015), and word and sentence vectors (Franco-Salvador et al., 2015). It is worth mentioning that most of the deep learning-based approaches performed poorly in the DSL shared task when compared to traditional classifiers, with one exception, the character-level CNN used by the MITSLS team (Belinkov and Glass, 2016).

3 Methodology

This section presents the corpus and the experiments performed in the DSL 2017 task by the system.

3.1 Corpus

The corpus compiled for the DSL 2017 shared task is composed of excerpts of journalistic texts, and it is divided into training, development, and test subsets. For this work, the training and development subsets were joined to train the system. The corpus is balanced in terms of sentences per language. For each of the 14 languages (classes) considered in the task, the training set consists of 18,000 sentences and the development set of 2,000 sentences. The entire corpus contains 252,000 sentences for training, 28,000 for development, and 14,000 for testing (1,000 sentences per language/variety).

As mentioned above, languages are grouped by similarity or common origin. Six groups are considered (each language code is indicated in brackets): (A) Bosnian (`bs`), Croatian (`hr`), and Serbian (`sr`); (B) Malay (`my`) and Indonesian (`id`); (C) Persian (`fa-IR`) and Dari (`fa-AF`); (D) Canadian (`fr-CA`) and Hexagonal French (`fr-FR`); (E) Brazilian (`pt-BR`) and European Portuguese (`pt-PT`); and (F) Argentinian (`es-AR`), Peruvian (`es-PE`), and Peninsular Spanish (`es-ES`).

3.2 Experimental settings

Let us now move to describe the experimental settings for the three runs submitted to the competition. Table 1 summarizes the experimental settings presented below.

For runs 1 and 2, a two-step classification approach was examined, since it has previously been proved to be a useful strategy for this task (Goutte et al., 2014; Goutte et al., 2015). In this approach, the language group is predicted first, and then the closely-related languages are discriminated within the group. This approach was compared against a single-step classification (run 3), where all the 14 languages of the corpus are discriminated, irrespective of their grouping.

The performance of two machine-learning classifiers was compared using their WEKA's (Witten et al., 2016) implementation with default parameters: Support Vector Machines (SVM) and Multinomial Naive Bayes (MNB). These classification algorithms are considered among the best for text categorization tasks (Kibriya et al., 2005; Zampieri et al., 2015). Moreover, SVM was the classifier of choice of the majority of the teams in the previous edition of the DSL shared task (Malmasi et al., 2016).

In the two-step approach (runs 1 and 2), the first step is the language group discrimination, which was performed using SVM classifier. Due to time constraints, in the second step (language/variety discrimination within a group) these two runs were set differently. In run 1, different algorithms were used for different language groups: SVM was used for groups B (Malay and Indonesian), C (Persian and Dari), and D (Canadian and Hexagonal French); while MNB was used for groups A (Bosnian, Croatian, and Serbian), E (Brazilian and European Portuguese), and F (Argentine, Peninsular, and Peruvian Spanish). In run 2, all language groups were discriminated using only MNB. For language group classification (runs 1 and 2), we increased the number of instances for training the classifier by duplicating and in some cases triplicating the training instances. In the single-step approach (run 3), only MNB was used to discriminate between the 14 languages (without group classification).

The performance of different feature sets was examined, using term frequency (*tf*) weighting scheme. Only features with $tf \geq 5$ were selected, that is, only those features that occur at least five times in the training corpus. The features used are the following: (i) unigrams of words, (ii) untyped (traditional) character n-grams, and (iii) typed character n-grams, that is, character n-grams classified into the categories introduced by Sapkota *et al.* (2015). The authors defined 10 different character n-gram categories based on affixes, words, and punctuation. In more detail, there are 3 main types, and each one has sub-categories as explained below:

- **Affix character n-grams**

  prefix An n-gram that covers the first n characters of a word that is at least n + 1 characters long.

  suffix An n-gram that covers the last n characters of a word that is at least n + 1 characters long.

  space-prefix An n-gram that begins with a space and that does not contain any punctuation mark.

Experimental settings		Run 1	Run 2	Run 3
Approach		two-step (6 groups; 14 languages)		single-step (14 languages)
ML algorithm (WEKA implementation, default parameters)	1^{st} step	SVM	SVM	MNB
	2^{nd} step	SVM: groups B, C, and D MNB: groups A, E, and F	MNB: all groups	
Features		untyped char. n-grams (n = 3–5), typed char. 3-grams (Sapkota et al., 2015), word unigrams.	same as run 1	same as run 1
Settings		*tf* weighting scheme; *frq*≥5	same as run 1	same as run 1

Table 1: Experimental settings in the three runs of the system.

space-suffix An n-gram that ends with a space, that does not contain any punctuation mark, and whose first character is not a space.

- **Word character n-grams**

whole-word An n-gram that encompasses all the characters of a word, and that is exactly n characters long.

mid-word An n-gram that contains n characters of a word that is at least n + 2 characters long, and that does not include neither the first nor the last character of the word.

multi-word An n-gram that spans multiple words, identified by the presence of a space in the middle of the n-gram.

- **Punctuation character n-grams**

beg-punct An n-gram whose first character is a punctuation mark, but the middle characters are not.

mid-punct An n-gram whose middle character is a punctuation mark (for n =3).

end-punct An n-gram whose last character is punctuation mark, but the first and the middle characters are not.

In this approach, instances of the same untyped n-gram may refer to different typed n-gram features. For example, in the phrase *the mother*, the first instance of the 3-gram *the* is assigned to a *whole-word* category and the second instance to a *mid-word* category. As an example, let us consider the following sample sentence:

(1) *Ana said, "Tom will fix it tomorrow."*

The character n-grams (n = 3) for the sample sentence (1) for each of the categories are shown in Table 2.

SC	Category	*N*-grams
affix	*prefix*	sai wil tom
	suffix	aid ill row
	space-prefix	_sa _wi _fi _it _to
	space-suffix	na_ om_ ll_ ix_ it_
word	*whole-word*	Ana Tom fix
	mid-word	omo mor orr rro
	multi-word	a_s m_w l_f x_i t_t
punct	*beg-punct*	,_“ “To
	mid-punct *	_,_ _“_ _._ _”_
	end-punct	id, ow.

* In our approach, punctuation marks are separated from adjacent words and from each other by space for this category. This enables to capture their frequency.

Table 2: Character 3-grams per category for the sample sentence (1) after applying the algorithm by Sapkota *et al.* (2015).

Different lengths of character n-grams were tested. Besides, and following previous studies (Malmasi and Dras, 2015), we examine whether the performance of the proposed models could be enhanced when combining different feature sets, i.e., typed and untyped character n-grams and words. In all the runs, the combination of untyped character n-grams with n from 3 to 5, typed character 3-grams, and words was selected for the final submission.

Finally, several authors (Franco-Salvador et al., 2015; Jauhiainen et al., 2016) have mentioned using some pre-processing prior to the feature extraction for the DSL shared task. This often involves removing the distinction between upper- and lowercase characters, number simplification (reducing all digits to a single one) or removal of punctuation. In the previous VarDial edition, named entities were also replaced by a conventional string. Lastly, pre-processing has proved

to be a useful strategy for several other classification tasks, including Author Profiling in social media texts (Gómez-Adorno et al., 2016a; Gómez-Adorno et al., 2016b), cross-genre Author Profiling (Markov et al., 2016), and similarity detection between programming languages (Sidorov et al., 2016). Though several experiments have been conducted using different pre-processing techniques, these failed to improve the results. Hence all pre-processing techniques have been dropped altogether, and those experiments are not reported here. Still, this can indicate that pre-processing removes features relevant to the DSL task.

The appropriate tuning of feature set size has proved to be important in other NLP tasks, such as Authorship Attribution (Stamatatos, 2013), and Author Profiling (Markov et al., 2016). In this work, an attempt was made to select the most appropriate frequency threshold based on a grid search. In more detail, the following frequency threshold (*frq*) values were examined: $frq = 5$, 10, 20, 50, and 100. Other experiments were also carried out by cutting out the most frequently occurring features in the training corpus, namely by discarding the 100 most frequent words. This strategy has proved to be helpful in other classification tasks, such as Author Profiling (Markov et al., 2016). However, in the DSL task discarding the most frequent features did not lead to improvements in accuracy. This result indicates that the most frequent words, which are stop-words for the most part, are important for DSL.

4 Experimental Results

Table 3 shows the final ranking of all the participating teams on the closed track of the DSL shared task. Except for the last system, results of all the participants are relatively similar, their accuracy ranging from 0.9274 (CECL) to 0.8894 (BAYESLINE), that is a difference of 0.038. The best submitted run (run 2) of the CIC_UALG team was ranked 6th among the 11 participants. However, the difference in accuracy from the 1th place is only 0.0128.

Next, the results of the three runs on the DSL 2017 test set are presented in Table 4. Firstly, the results of run 3 (single-step, 14 languages and no language group classification, using MNB) are slightly worse than those for runs 1 and 2 (0.0052 and 0.0077, respectively). This seems to confirm the validity of the two-step approach. Secondly,

Team	Rank	Accuracy
CECL	1	0.9274
MM_LCT	2	0.9254
XAC_BAYESLINE	3	0.9247
TUBASFS	4	0.9249
GAUGE	5	0.9165
CIC_UALG	**6**	**0.9146**
SUKI	7	0.9099
TIMEFLOW	8	0.9076
CITIUS_IXA_IMAXIN	9	0.9030
BAYESLINE	10	0.8894
DEEPCYBERNET	11	0.2046

Table 3: Final ranking for the closed track of the DSL shared task.

results of run 2 (two-step classification approach using SVM for groups and MNB for languages) slightly outperformed those of run 1 (similar setting to those of run 2, but using SVM or MNB depending on language group). This behavior was the opposite of the one seen in the experiments conducted on the development set, where the best results were achieved using an SVM classifier for both group and language classification. Since time constraints precluded repeating in run 1 test set (mixed SVM/MNB in the second step) exactly the experimental settings adopted for the development set (only SVM in both classification steps), it remains to be seen whether such scenario would change the results, and by how much.

Group classification is extremely important, since a model is unable to recover from mistakes made at the group prediction step. Table 5 shows the performance of run 2 for the language group classification. The overall results for all the language groups are very high and are in line with the experiments on the development set, where similar results were achieved.

As one can see from Table 6, the results for language classification are lower than those for group classification. The most challenging languages are the ones in groups A and F, where the average precision is 0.85 and 0.88, respectively. In group A, the Bosnian language showed a precision of 0.79, which makes it the most difficult language to identify when compared with Serbian and Croatian. Another interesting result emerges from the results concerning the Spanish language (group F), which also show a wide variation in the performance of the classifiers. This may be due to the (relatively)

Run	Accuracy	F1 (micro)	F1 (macro)	F1 (weighted)
run 1	0.9121	0.9121	0.9121	0.9121
run 2	**0.9146**	**0.9146**	**0.9146**	**0.9146**
run 3	0.9069	0.9069	0.9068	0.9068

Table 4: Results in terms of accuracy and F1 measures for the three submitted runs on the test set.

Language	Precision	Recall	F1-score
Group A	0.9980	0.9990	0.9985
Group B	0.9995	0.9975	0.9985
Group C	1.0000	0.9995	0.9997
Group D	0.9965	0.9995	0.9980
Group E	0.9940	0.9970	0.9955
Group F	0.9987	0.9953	0.9970

Table 5: Performance of run 2 per group of languages.

autonomous evolution of the American varieties not having followed the innovations of the Peninsular variety. Notice that, in comparison, the system shows a much more similar behavior when distinguishing the two Portuguese varieties, whose historic drift is also very evident.

Language	Precision	Recall	F1-score
hr	0.87	0.83	0.85
bs	0.79	0.79	0.79
sr	0.88	0.93	0.90
id	0.99	0.98	0.98
my	0.98	0.98	0.98
fa-af	0.97	0.94	0.95
fa-ir	0.94	0.97	0.96
fr-ca	0.95	0.93	0.94
fr-fr	0.92	0.95	0.94
pt-br	0.93	0.95	0.94
pt-pt	0.95	0.93	0.94
es-ar	0.87	0.86	0.86
es-es	0.85	0.88	0.87
es-pe	0.92	0.90	0.91

Table 6: Performance of run 2 per language.

The confusion matrix for our best run (run 2) in the closed DSL task is shown in Figure 1. The greatest confusion is in the Bosnian-Croatian-Serbian group, followed by the Spanish and Portuguese dialect groups. Bosnian is the most difficult language for identification among all the 14 classes.

5 Typed N-grams

A new type of features was introduced for the DSL task, typed character n-grams. Table 7 shows the different feature combinations experimented for the first step (language group) classification task, the number of features (N) considered in each experiment and the corresponding accuracy (Acc. (%)). For lack of space, only the experiments with typed 3-grams (and one experiment with 4-grams), using a frequency threshold of *frq*=20 and the SVM algorithm are shown here.

Words	Untyped 3-grams	Typed 3-grams	Untyped 4-grams	Typed 4-grams	Untyped 5-grams	Untyped 6-grams	N	Acc. (%)
✓							40,525	99.5607
	✓						36,626	99.7893
		✓					43,390	99.7929
	✓	✓					80,016	99.8071
✓	✓	✓					120,541	99.8214
✓	✓	✓	✓				240,322	99.8214
✓	✓	✓	✓		✓		**493,075**	**99.8250**
✓	✓	✓	✓		✓	✓	847,782	99.8250
✓	✓	✓	✓	✓	✓	✓	956,295	99.8071

Table 7: Results from different feature combinations on the language group classification step over the development set.

It is possible to observe that the basic bag-of-words approach (*Words*) already performs at a very reasonable level (99.5607%), but also that this result was always outperformed in all the other experiments where n-gram features were added.

Secondly, there is a slight increase (0.0036) in the performance when the typed 3-grams are used, instead of just the traditional, untyped 3-grams. The size of the feature set, however, also increases. Combining typed and untyped 3-grams improves the results further (0.142), while combining words and both kinds of n-grams provides an even better accuracy (99.8214%), a result 0.2607 above the simple, bag-of-words approach.

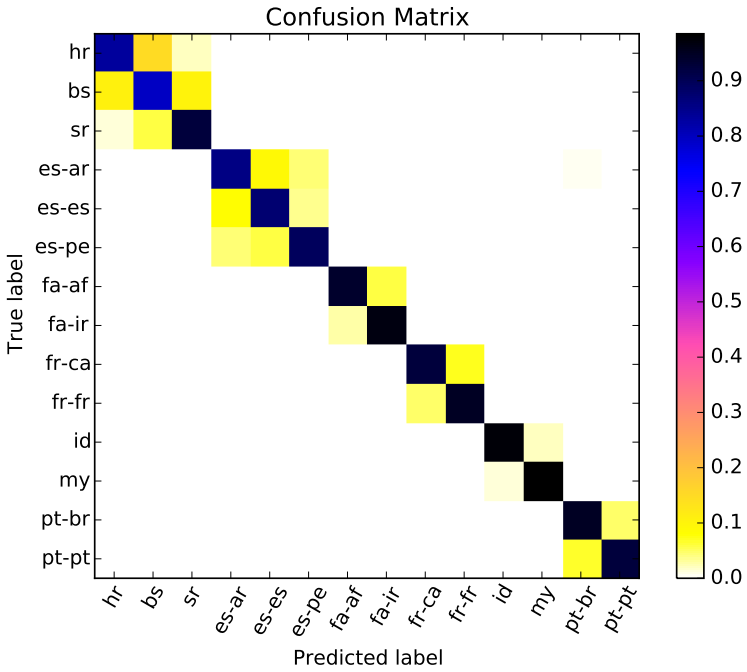

Figure 1: Confusion matrix of run 2.

In the next experiments, we successively added larger untyped n-grams to the feature set, with n from 4 to 6. Naturally, the size of the feature set increases significantly. Adding larger n-grams increased the results up to 99.8250% accuracy, but it is noteworthy that the untyped 6-grams did not improve the results above those already obtained with the untyped 5-grams, while the feature set increases 1.72 times.

Finally, a new set of typed 4-grams was added to the previous experimental settings. This, however, hindered the results, producing the same accuracy as just combining typed and untyped 3-grams. Notice that size of the feature set is approximately 12 times larger than that experiment.

As far as the language classification within language groups is concerned, experiments were carried out comparing the use of typed against untyped n-grams on the development set. Typed n-grams systematically outperformed the untyped ones. Moreover, different feature combinations were also tested for language classification; however, none of them was able to outperform the feature combination selected for the language group classification (typed 3-grams, untyped n-grams (n = 3–5), and words), and therefore, this combination was also selected for discriminating between the languages within the group.

6 Conclusions

This paper presented the description of the three runs submitted by the CIC_UALG team to the Discriminating between Similar Languages (DSL) shared task at the VarDial 2017 Workshop. The best performance was obtained by run 2, which achieved an accuracy of 0.9146 (6$^{\text{th}}$ place out of 11). This run implements a two-step classification approach, predicting first the group of languages and then discriminating the languages within the group.

Typed character n-grams was a new type of features that had been introduced in the DSL task for the first time. It was found during the preliminary experiments (on the development set) that these features improve the classification accuracy when used in combination with other types of features such as word unigrams and untyped n-grams. It was demonstrated that having increasingly larger typed or untyped n-grams can only improve results up to a certain point, and then performance deteriorates. A careful selection of feature combinations is thus required to obtain optimal results while controlling the increase in the size of the feature set, which can become computationally too costly.

One of the directions for future work would be to conduct experiments using doc2vec-based (distributed) feature representation, which has proved to provide good results for DSL (Franco-Salvador et al., 2015) and other NLP tasks, such as Authorship Attribution (Posadas-Durán et al., 2016) and Author Profiling (Markov et al., 2017), among others. Moreover, classifier ensembles will be examined, since it has been demonstrated that they are efficient for DSL (Malmasi and Dras, 2015), as well as for different real-word problems (Oza and Tumer, 2008).

Acknowledgments

This work was partially supported by the Mexican Government (Conacyt projects 240844 and 20161958, SIP-IPN 20151406, 20161947, 20161958, 20151589, 20162204, and 20162064, SNI, COFAA-IPN) and by the Portuguese Government, through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

References

Yonatan Belinkov and James Glass. 2016. A character-level convolutional neural network for distinguishing similar languages and dialects. In *Proceedings of the 3rd Workshop on NLP for Similar Languages, Varieties and Dialects*, VarDial '16, pages 145–150.

Victoria Bobicev. 2015. Discriminating between similar languages using PPM. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, LT4VarDial '15, pages 59–65.

Çağrı Çöltekin and Taraka Rama. 2016. Discriminating similar languages: experiments with linear SVMs and neural networks. In *Proceedings of the 3rd Workshop on NLP for Similar Languages, Varieties and Dialects*, VarDial'16, pages 15–24.

Marc Franco-Salvador, Paolo Rosso, and Francisco Rangel. 2015. Distributed representations of words and documents for discriminating similar languages. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, LT4VarDial '15, pages 11–16.

Helena Gómez-Adorno, Ilia Markov, Grigori Sidorov, Juan-Pablo Posadas-Durán, and Carolina Fócil-Arias. 2016a. Compilación de un lexicón de redes sociales para la identificación de perfiles de autor. *Research in Computing Science*, 115:19–27.

Helena Gómez-Adorno, Ilia Markov, Grigori Sidorov, Juan-Pablo Posadas-Durán, Miguel A. Sanchez-Perez, and Liliana Chanona-Hernandez. 2016b. Improving feature representation based on a neural network for author profiling in social media texts. *Computational Intelligence and Neuroscience*, 2016:13 pages.

Cyril Goutte, Serge Léger, and Marine Carpuat. 2014. The NRC system for discriminating similar languages. In *Proceedings of the Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, VarDial '14, pages 139–145.

Cyril Goutte, Serge Léger, and Marine Carpuat. 2015. Experiments in discriminating similar languages. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, LT4VarDial '15, pages 78–84.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating similar languages: Evaluations and explorations. In *Proceedings of the Language Resources and Evaluation (LREC)*, pages 1800–1807, Portoroz, Slovenia.

Radu Ionescu and Marius Popescu. 2016. UnibucKernel: An approach for Arabic dialect identification based on multiple string kernels. In *Proceedings of the 3rd Workshop on NLP for Similar Languages, Varieties and Dialects*, VarDial '16, pages 135–144.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2016. HeLI, a word-based backoff method for language identification. In *Proceedings of the 3rd Workshop on NLP for Similar Languages, Varieties and Dialects*, VarDial'16, pages 153–162.

Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. 2005. Multinomial naive Bayes for text categorization revisited. In *Proceedings of the 17th Australian joint conference on Advances in Artificial Intelligence*, AI '04, pages 488–499.

Marco Lui and Timothy Baldwin. 2011. Cross-domain feature selection for language identification. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, IJCBLP '17, pages 553–561.

Suraj Maharjan and Thamar Solorio. 2015. Using wide range of features for author profiling. In *CLEF (Working Notes)*, volume 1391.

Shervin Malmasi and Mark Dras. 2015. Language identification using classifier ensembles. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, LT4VarDial'15, pages 35–43.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between similar languages and arabic dialect identification: A report on the third DSL shared task. In *Proceedings of the*

3rd Workshop on Language Technology for Closely Related Languages, Varieties and Dialects, VarDial '16, pages 1–14.

Ilia Markov, Helena Gómez-Adorno, Grigori Sidorov, and Alexander Gelbukh. 2016. Adapting cross-genre author profiling to language and corpus. In *Working Notes Papers of the CLEF 2016 Evaluation Labs*, volume 1609 of *CEUR Workshop Proceedings*, pages 947–955. CLEF and CEUR-WS.org.

Ilia Markov, Helena Gómez-Adorno, Juan-Pablo Posadas-Durán, Grigori Sidorov, and Alexander Gelbukh. 2017. Author profiling with doc2vec neural network-based document embeddings. In *Proceedings of the 15th Mexican International Conference on Artificial Intelligence*, volume 10062 of *MICAI '16*. LNAI, Springer.

Nikunj Oza and Kagan Tumer. 2008. Classifier ensembles: Select real-world applications. *Information Fusion*, 9(1):4–20.

Juan-Pablo Posadas-Durán, Helena Gómez-Adorno, Grigori Sidorov, Ildar Batyrshin, David Pinto, and Liliana Chanona-Hernández. 2016. Application of the distributed document representation in the authorship attribution task for small corpora. *Soft Computing*, 21:627–639.

Upendra Sapkota, Steven Bethard, Manuel Montes-y-Gómez, and Thamar Solorio. 2015. Not all character n-grams are created equal: A study in authorship attribution. In *Proceedings of the 2015 Annual Conference of the North American Chapter of the ACL: Human Language Technologies*, NAACL-HLT '15, pages 93–102.

Grigori Sidorov, Martín Ibarra Romero, Ilia Markov, Rafael Guzman-Cabrera, Liliana Chanona-Hernández, and Francisco Velásquez. 2016. Detección automática de similitud entre programas del lenguaje de programación Karel basada en técnicas de procesamiento de lenguaje natural. *Computación y Sistemas*, 20(2):279–288.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Gohneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the 1th Workshop on Computational Approaches to Code Switching*, pages 62–72.

Efstathios Stamatatos. 2013. On the robustness of authorship attribution based on character n-gram features. *Journal of Law & Policy*, 21(2):427–439.

Liling Tan, Marcos Zampieri, Nikola Ljubešic, and Jörg Tiedemann. 2014. Merging comparable data sources for the discrimination of similar languages: The dsl corpus collection. In *Proceedings of the 7th Workshop on Building and Using Comparable Corpora*, BUCC '14, pages 11–15.

Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. 2016. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 4th edition.

Marcos Zampieri, Liling Tan, Nikola Ljubešic, and Jörg Tiedemann. 2014. A report on the DSL shared task 2014. In *Proceedings of the Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, VarDial '14, pages 58–67.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the DSL shared task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, LT4VarDial '15, pages 1–9.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the vardial evaluation campaign 2017. In *Proceedings of the 4th Workshop on NLP for Similar Languages, Varieties and Dialects*, VarDial '17.

Ayah Zirikly, Bart Desmet, and Mona Diab. 2016. The GW/LT3 VarDial 2016 shared task system for dialects and similar languages detection. In *Proceedings of the 3rd Workshop on NLP for Similar Languages, Varieties and Dialects*, VarDial'16, pages 33–41.