Vanilla Classifiers for Distinguishing between Similar Languages

Alina Maria Ciobanu, Sergiu Nisioi, Liviu P. Dinu
Solomon Marcus Center for Computational Linguistics,
Faculty of Mathematics and Computer Science,
University of Bucharest
alina.ciobanu@my.fmi.unibuc.ro,
sergiu.nisioi@gmail.com,
ldinu@fmi.unibuc.ro

Abstract

In this paper we describe the submission of the UniBuc-NLP team for the Discriminating between Similar Languages Shared Task, DSL 2016. We present and analyze the results we obtained in the closed track of sub-task 1 (Similar languages and language varieties) and sub-task 2 (Arabic dialects). For sub-task 1 we used a logistic regression classifier with tf-idf feature weighting and for sub-task 2 a character-based string kernel with an SVM classifier. Our results show that good accuracy scores can be obtained with limited feature and model engineering. While certain limitations are to be acknowledged, our approach worked surprisingly well for out-of-domain, social media data, with 0.898 accuracy (3rd place) for dataset B1 and 0.838 accuracy (4th place) for dataset B2.

1 Introduction

Automatic language identification is the task of determining the language in which a piece of text is written using computational methods. In today's context of multilingualism, and given the rapid development of the online repositories of cross-language information, language identification is an essential task for many downstream applications (such as cross-language information retrieval or question answering), to route the documents to the appropriate NLP systems, based on their language.

Although language identification has been intensively studied in the recent period, there are still questions to be answered. Language identification is still a challenging research problem for very similar languages and language varieties, for very short pieces of text, such as tweets, or for documents involving code-switching (the practice of mixing more languages within a single communication).

The DSL 2016 shared task (Malmasi et al., 2016) tackles two interesting aspects of language identification: similar language and language varieties (with in-domain and out-of-domain – social media data – test sets) and Arabic dialects. In this paper we present the submission of the UniBuc-NLP team for the closed track (using only the training data provided by the organizers) of both sub-tasks.

2 Related Work

Most approaches to language identification are based on character n-grams. Dunning (1994) was one of the very first who used them. He proposed a statistical method for language identification based on Markov models to compute the likelihood of the character n-grams. Ever since, character n-grams have been employed to discriminate between a wide variety of closely related languages and dialects. Maier and Gómez-Rodríguez (2014) performed language classification on tweets for Spanish varieties, with character n-grams as features and using the country of the speaker to identify the variety. Trieschnigg et al. (2012) discriminated between Dutch dialects (and several other languages) using a large collection of folktales. They compared several approaches to language identification and reported good results when using the method of Cavnar and Trenkle (1994), based on character n-grams. Sadat et al. (2014) performed language identification on Arabic dialects using social media texts. They obtained better results with Naive Bayes and n-gram features (2-grams) than with a character n-gram Markov model for

Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects,
pages 235–242, Osaka, Japan, December 12 2016.

most of the Arabic dialects. Gottron and Lipka (2010) conducted a comparative experiment of classification methods for language identification in short texts, discriminating between languages from various language families and using n-gram features. Their results show that Naive Bayes classifier performs best and that errors occur for languages from the same family, reinforcing the hypothesis that language identification is more difficult for very similar languages.

Word n-grams have also proven effective for discriminating between languages and language varieties. Malmasi and Dras (2015) achieved the best performance in the closed track of the DSL 2015 shared task, experimenting with classifier ensembles trained on character and word n-gram features. Goutte and Leger (2015) obtained a very good performance in the same competition using statistical classifiers and employing a combination of character and word n-grams as features. Zampieri and Gebre (2012) made use of a character n-gram model and a word n-gram language model to discriminate between two varieties of Portuguese. They reported the highest accuracy when using character 4-grams and reached the conclusion that orthographic and lexical differences between the two varieties have more discriminative power than lexico-syntactic differences.

Other features, such as exclusive words, the format of the numbers (Ranaivo-Malancon, 2006), blacklists (Tiedemann and Ljubesic, 2012), syllable n-grams (Maier and Gómez-Rodríguez, 2014) or skip-grams have been employed and shown useful for this task.

3 Data

The organizers released two training datasets for the 2016 DSL shared task: a dataset of similar languages and language varieties (for sub-task 1) and a dataset of Arabic dialects (for sub-task 2).

The dataset for sub-task 1 is a new version of the DSL Corpus Collection (Tan et al., 2014). It contains instances written in the following languages and language varieties (organized by groups of similarity):

Language	Lang. code	Group code	Avg. sent. lenth	Avg. word length
Bosnian	bs	bs-hr-sr	31.38	5.21
Croatian	hr		37.30	5.30
Serbian	sr		34.28	5.09
Indonesian	id	id-my	34.34	5.84
Malay	my		26.01	5.91
Portuguese (Brazil)	pt-BR	pt	39.94	4.90
Portuguese (Portugal)	pt-PT		36.70	4.92
Spanish (Argentina)	es-AR	es	41.70	4.98
Spanish (Mexico)	es-MX		30.96	4.78
Spanish (Spain)	es-ES		45.06	4.84
French (France)	fr-FR	fr	37.13	4.69
French (Canada)	fr-CA		30.20	4.69

Table 1: Statistics for the dataset of similar languages and language varieties (sub-task 1).

The dataset consists of 20,000 instances (18,000 for training and 2,000 for development) in each language or language variety, extracted from journalistic texts. In Table 1 we report several statistics for this dataset. The average sentence length varies from 26.01 (for Malay) to 45.06 (for the Spanish variety used in Spain). We observe a high variance for the average sentence length within some of the language groups (the difference between the average sentence length of Indonesian and Malay is $\sim$8, and between the average sentence length of the Spanish variety spoken in Spain and the one spoken in Mexico is $\sim$14). The average word length varies from 4.69 (for both versions of French) to 5.91 (for Malay), with a low variance within groups.

Comparing these statistics with those extracted from the sub-task 1 test sets, we notice that while the

average sentence length values for test set A are similar to those of the training set, for test sets B1 and B2 – social media data – sentences are significantly shorter, as expected, ranging from an average of 11.33 for Portuguese (Brazil) to an average of 13.39 for Serbian. The average word length values for B1 and B2 are also smaller than those for test set A and the training set, but the differences are not as significant as the differences regarding the length of the sentences.

The dataset for sub-task 2 contains automatic speech recognition transcripts (Ali et al., 2016) written in the following Arabic dialects: Egyptian, Gulf, Levantine, North-African, and Modern Standard Arabic. In Table 2 we report several statistics for this dataset. The average sentence length ranges from 35.41 (for North-African) to 60.57 (for Egyptian). All the Arabic dialects have the average word length lower than 4.

Dialect	Dialect code	# instances	Avg. sent. lenth	Avg. word length
Egyptian	EGY	1,578	60.57	3.65
Gulf	GLF	1,672	43.21	3.64
Levantine	LAV	1,758	42.01	3.63
North-African	NOR	1,612	35.41	3.74
Modern Standard Arabic	MSA	999	56.94	3.80

Table 2: Statistics for the dataset of Arabic dialects (sub-task 2).

4 Our Approach

In this section we describe and analyze the methods we used for discriminating between similar languages, language varieties and dialects. We used standard linear classifiers with basic n-grams features.[1]

4.1 Classifiers

Logistic Regression

For sub-task 1 we used a logistic regression classifier with word unigrams and bigrams as features. The features are tf-idf (Salton and Buckley, 1988) weighted and we keep only the features that occur at least 3 times in the training set. We use the L_2 distance for term vectors and the default regularization constant $C = 1$ without performing any grid search for best parameters. We use the wrapper of the scikit learn Python library (Pedregosa et al., 2011) over the Liblinear logistic regression implementation (Fan et al., 2008). The main advantages of this model are its simplicity and training speed.

SVM + String Kernel

On the Arabic dataset we decided to use string a kernel based on character n-grams, since the text is obtained through ASR systems and most certainly the transcript contains errors. Character n-grams are able to cover sub-parts of words and can theoretically increase the overall classification accuracy, especially in a language recognition task. We used a string kernel in combination with a support vector machine classifier. A kernel function can be used either to embed the data in a higher dimensional space to achieve linear separability, or to replace the dot product between vectors with values that are more appropriate for the data used. Previous studies on text classification revealed that character n-gram-based string kernels can be effective tools for authorship attribution, native language identification or plagiarism detection (Grozea and Popescu, 2010).

The kernel we propose is computed by summing the number of common character n-grams between two examples, where n varies between 2 and 7. Formally, given an alphabet A, we define the mapping function $\Phi_n : \mathcal{D} \rightarrow \{0,1\}^{Q_n}$ for an example $e \in \mathcal{C}$ in the corpus to be the vector of all the binary values of existence of the n-gram g in the document:

$$\Phi_n(e) = [\phi_g(e)]_{g \in A^n}$$

[1]The source code to reproduce our results is available at `https://gitlab.com/nlp-unibuc/dsl2016-code/`.

The function $\phi_g(e) = 1$ if the n-gram g is in the example e and equal to zero otherwise. Computationally, Q_n depends on all the possible character n-grams between two examples at certain instance.

The corresponding Gram matrix K of size $|\mathcal{C}| \times |\mathcal{C}|$ has the following elements:

$$K_{ij} = \sum_{n=2}^{n \leq 7} < \Phi_n(e_i)\Phi_n(e_j) >$$

The Gram matrix is then normalized to the $[0, 1]$ interval:

$$K_{ij} = \frac{K_{ij}}{\sqrt{K_{ii}K_{jj}}} \tag{1}$$

The kernel function, in our case, is computed between every pair of training and testing examples. This type of approach is less scalable for large amounts of data, which is the main reason for not applying this technique on sub-task 1. However, additional optimizations can be taken into consideration, such as using just the upper half of the symmetric Gram matrix, aggregating multiple kernels trained on sub-samples of the data or hashing techniques for faster computation. In our vanilla approach we did not make use of any of these techniques.

In practice, kernel methods over strings for text classification work remarkably well covering fine-grained similarities such as content, punctuation marks, affixes etc., however one important down-side of this method is usually the lack of linguistic features available within the classifier, making almost impossible to analyze from the Gram matrix the actual features that lead to good or bad results.

4.2 Experiments

Using the experimental setup previously described, we developed several systems for discriminating between similar languages and language varieties (sub-task 1) and between Arabic dialects (sub-task 2).

The organizers provided three test datasets, two for sub-task 1 and one for sub-task 2. In Table 3 we provide a brief characterization of the datasets:

Dataset	Description	Task	# instances
A	In-domain: newspaper texts	Sub-task 1	12,000
B1	Out of domain: social media data	Sub-task 1	500
B2	Out of domain: social media data	Sub-task 1	500
C	ASR texts from Arabic dialects	Sub-task 2	1,540

Table 3: Test datasets for DSL 2016.

Sub-task 1

Our two runs for sub-task 1 are as follows:

- **Run 1:** a one-level system. The first system consists of a single logistic regression classifier that predicts the language or language variety.
- **Run 2:** a two-level system. The second system consists of multiple logistic regression classifiers: we train a classifier to predict the language group ("inter-group classifier"), and one classifier for each language group ("intra-group classifier"), to predict the language or language variety within the group.

For the one-level system we obtained 0.8441 accuracy when evaluating on the development dataset. For the two-level system we obtained 0.9972 accuracy for the inter-group classifier, and the following values for the intra-group classifiers: 0.7510 for es, 0.8940 for fr, 0.9207 for pt, 0.7848 for bs-hr-sr, 0.9820 for id-my.

Test Set	Run	Accuracy	F1 (micro)	F1 (macro)	F1 (weighted)
A	Run 1	0.8624	0.8624	0.8620	0.8620
A	**Run 2**	**0.8648**	**0.8648**	**0.8643**	**0.8643**
B1	**Run 1**	**0.8980**	**0.8980**	**0.7474**	**0.8969**
B1	Run2	0.8940	0.8940	0.7429	0.8915
B2	Run1	0.8360	0.8360	0.5970	0.8358
B2	**Run2**	**0.8380**	**0.8380**	**0.5236**	**0.8378**

Table 4: The results of the UniBuc-NLP team for sub-task 1.

Test Set	Run	Accuracy	F1 (micro)	F1 (macro)	F1 (weighted)
C	run1	0.3948	0.3948	0.3891	0.3938
C	run2	0.4747	0.4747	0.4729	0.4732
C	**run3**	**0.4753**	**0.4753**	**0.4732**	**0.4742**

Table 5: The results of the UniBuc-NLP team for sub-task 2.

In Table 4 we report the results that we obtained for the test datasets. Our best results for each dataset are as follows: 0.8648 accuracy (11th place) for dataset A, 0.8980 accuracy (3rd place) for dataset B1 and 0.8380 accuracy (4th place) for dataset B2. For two of the three datasets (A, B2), the two-level system obtained better results than the one-level system. However, our highest accuracy (0.8990) was obtained by the one-level system for dataset B1.

Lang. code	Top 10 informative features
bs	povrije ,fbih, rs, poslije, km, prenosi, je, sarajevo, bh, bih
hr	tko, hdz, je, hrvatska, milijuna, u, te, kuna, tijekom, s
sr	evra, deo, srbije, predsednik, dve, vreme, gde, da, pre, posle
id	tim, tak, indonesia, mengatakan, di, bahwa, saat, dari, karena, bisa
my	ialah, encik, turut, apabila, selepas, boleh, berkata, daripada, beliau, kerana
pt-BR	para, ela, voce, do, em, brasil, r, e, o, ele
pt-PT	acores, o seu, a sua, numa, equipa, num, e, euros, a, portugal
es-AR	productores, empresas, ar, de rosario, el, santa fe, de, y, argentina, rosario
es-MX	mexicano, gadafi, mil, el, mexico, dijo, en, la, que, de
es-ES	alicante, murcia, del, ayer, han, la, y, euros, el, ha
fr-FR	d, paris, euros, est, et, les, le, l, france, vous
fr-CA	des, dit, de, de montreal, mme, quebecois, m, canada, montreal, quebec

Table 6: The most informative features for the one-level system for sub-task 1.

In Tables 6 and 7 we report the most informative features for each class. With few exceptions, most of the informative features are unigrams. While for the language classifiers many of these features are named entities (such as references to geographical regions or names of persons), as expected, for the language group classifier (Table 7a) the situation is different: mostly very short words prove to have high discriminative power. Among others, we identified definite and indefinite articles – "los" (es), "le" (fr) – and functional words – "nao" (pt), "dalam" (id-my) – ranked among the most informative features.

Despite the fact that quite many of the top features are named entities, which could a suggest a topic bias in classification, our systems obtain a good performance on out-of-domain data, ranking 3rd and 4th on the social media datasets.

Both our systems outperform significantly a random baseline that obtains 0.0883 F1 score for dataset A and 0.20 for datasets B1 and B2.

Group code	Top 10 informative features
bs-hr-sr	da, ce, iz, su, od, na, za, u, i, je
id-my	dari, pada, dalam, ini, untuk, dengan, itu, di, yang, dan
pt	nao, a, com, um, as, os, em, do, o, e
es	una, las, con, de, la, del, en, los, el, y
fr	au, une, pour, d, du, l, des, les, et, le

(a) Level 1: language groups.

Lang. code	Top 10 informative features
bs	sarajeva,sarajevu, fbih, rs, poslije, prenosi, km, sarajevo, bh, bih
hr	hrvatska, tisuca, hdz, tko, milijuna, te, no, kuna, tijekom, s
sr	evra, deo, srbije, predsednik, dve, gde, vreme, da, pre, posle
id	harus, tak, indonesia, tim, mengatakan, bahwa, dari, saat, bisa, karena
my	encik, turut, bahawa, apabila, selepas, boleh, berkata, daripada, beliau, kerana
pt-BR	eles, equipe, voce, sao paulo, federal, ela, em um, brasil, r, ele
pt-PT	numa, lisboa, acores, num, o seu, a sua, este, equipa, euros, portugal
es-AR	provincial, produccion, productores, empresas, mercado, empresa, santa fe, de rosario, argentina, rosario
es-MX	pri, de mexico, japon, pues, gadafi, mexicano, libia, mil, dijo, mexico
es-ES	ayuntamiento, espana, y a, murcia, alicante, han, cantabria, ayer, euros, ha
fr-FR	est, l, sarkozy, 2, francais, paris, 1, euros, vous, france
fr-CA	canadiens, ottawa, harper, m, du quebec, de montreal, quebecois, canada, montreal, quebec

(b) Level 2: languages.

Table 7: The most informative features for the two-level system for sub-task 2.

Sub-task 2

Our three runs for sub-task 2 are as follows:

- **Run 1**: SVM + string kernel with n-gram size n $\in$ { 2,...,5 }.
- **Run 2**: SVM + string kernel with n-gram size n $\in$ { 2,...,6 }.
- **Run 3**: SVM + string kernel with n-gram size n $\in$ { 2,...,7 }.

In Table 5 we report the results that we obtained for the test dataset. As expected, the accuracy of the system increases as the range of n-grams becomes wider. Our best result for sub-task 2 is 0.4753 accuracy (8^{th} place). In Figures 1, 2 and 3 we render the confusion matrices for the classification of the Arabic dialects. We observe a different behavior for the five classes, along the three runs: for EGY and LAV, the number of correctly classified instances is very similar over the three runs. For GLF there is a slight increase in correctly classified instances at run 2. For MSA the increase is significant (from 92 in run 1 to 190 – more than double – in run 2), and for NOR there is a certain decrease (from 180 in run 1 to 145 in run 2).

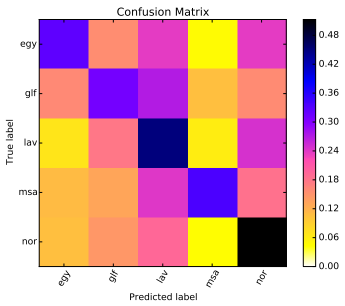

Figure 1: Run 1

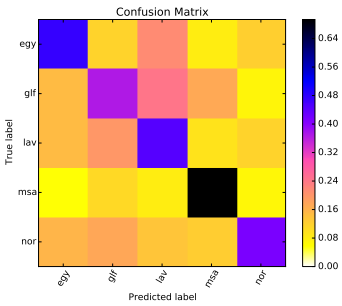

Figure 2: Run 2

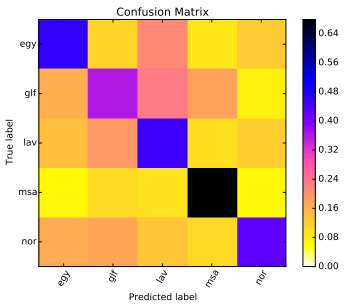

Figure 3: Run 3

5 Conclusions

In this paper we described the submission of the UniBuc-NLP team for the DSL 2016 shared task. We participated in the closed track of both sub-tasks (sub-task 1: similar languages and language varieties, sub-task 2: Arabic dialects), submitting a total of 5 runs (2 for sub-task 1 and 3 for sub-task 2). We used linear classification methods based on word and character n-gram features. For sub-task 1 we used a logistic regressions classifier with tf-idf feature weighting and for sub-task 2 an SVM classifier with a string kernel. Our best system obtains 89.80% accuracy for sub-task 1, dataset B1 (3rd place). Our results suggest that relatively good results may be obtained with plain vanilla linear classifiers, with no hyper-parameter optimization or special feature selection. When compared to other competitors in the shared task, our logistic regression results were at most 0.03% lower compared to the top score on sub-task 1, dataset A and among the top scoring for the datasets B1 and B2. On the Arabic dataset, the kernel method stands 0.04% from the first position and while additional parameters can improve the model, we believe the dataset created using ASR had a great impact on the results. To conclude, plain vanilla methods can be *good enough* to distinguish between similar languages, however we are still a long way from claiming this task *solved* and clearly more research is needed in this direction to create robust models that capture linguistic variation.

6 Acknowledgments

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI UEFISCDI, project number PN-III-P2-2.1-53BG/2016, within PNCDI III.

References

Ahmed Ali, Najim Dehak, Patrick Cardinal, Sameer Khurana, Sree Harsha Yella, James Glass, Peter Bell, and Steve Renals. 2016. Automatic dialect detection in arabic broadcast speech. In *Interspeech 2016*, pages 2934–2938.

William B. Cavnar and John M. Trenkle. 1994. N-Gram-Based Text Categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, SDAIR 1994*, pages 161–175.

Ted Dunning. 1994. Statistical Identification of Language. Technical report, Computing Research Laboratory, New Mexico State University.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874.

Thomas Gottron and Nedim Lipka. 2010. A Comparison of Language Identification Approaches on Short, Query-Style Texts. In *Proceedings of the 32nd European Conference on Advances in Information Retrieval, ECIR 2010*, pages 611–614.

Cyril Goutte and Serge Leger. 2015. Experiments in Discriminating Similar Languages. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, pages 78–84.

Cristian Grozea and Marius Popescu. 2010. Encoplot - performance in the second international plagiarism detection challenge - lab report for PAN at CLEF 2010. In *CLEF (Notebook Papers/LABs/Workshops)*, volume 1176 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Wolfgang Maier and Carlos Gómez-Rodríguez. 2014. Language Variety Identification in Spanish Tweets. In *Proceedings of the Workshop on Language Technology for Closely Related Languages and Language Variants, LT4CloseLang 2014*, pages 25–35.

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*, 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)*, Osaka, Japan.

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.

Fatiha Sadat, Farnazeh Kazemi, and Atefeh Farzindar. 2014. Automatic identification of arabic dialects in social media. In *Proceedings of the First International Workshop on Social Media Retrieval and Analysis, SoMeRA 2014*, pages 35–40.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing Management*, 24(5):513 – 523.

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)*, pages 11–15, Reykjavik, Iceland.

Jörg Tiedemann and Nikola Ljubesic. 2012. Efficient discrimination between closely related languages. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 2619–2634.

Dolf Trieschnigg, Djoerd Hiemstra, Mariët Theune, Franciska de Jong, and Theo Meder. 2012. An exploration of language identification techniques for the Dutch folktale database. In *Workshop on Adaptation of Language Resources and Tools for Processing Cultural Heritage, LREC 2012*, pages 47–51.

Marcos Zampieri and Binyam Gebrekidan Gebre. 2012. Automatic Identification of Language Varieties: The Case of Portuguese. In *Proceedings of the 11th Conference on Natural Language Processing, KONVENS 2012*, pages 233–237.