Advances in Ngram-based Discrimination of Similar Languages

Cyril Goutte
Multilingual Text Processing
National Research Council
Ottawa ON, Canada
Cyril.Goutte@gmail.com

Serge Léger
Human Computer Interaction
National Research Council
Moncton NB, Canada
Serge.Leger@nrc.ca

Abstract

We describe the systems entered by the National Research Council in the 2016 shared task on discriminating similar languages. Like previous years, we relied on character ngram features, and a combination of discriminative and generative statistical classifiers. We mostly investigated the influence of the amount of data on the performance, in the open task, and compared the two-stage approach (predicting language/group, then variant) to a flat approach. Results suggest that ngrams are still state-of-the-art for language and variant identification, that additional data has a small but decisive impact, and that the two-stage approach performs slightly better, everything else being kept equal, than the flat approach.

1 Introduction

We describe the systems submitted by the National Research Council Canada to the 2016 shared task on discriminating similar languages.

Discriminating similar languages and language variants is useful for several purposes. As the typical linguistic processing pipeline is tailored to a specific language and variant, it is important to have a reliable prediction of the language a text is written in, in order to use the appropriate linguistic tools. It may also be used for filtering data in order to build these specialized linguistic processing tools. In education, and language learning in particular, it may also be useful to identify precisely the variant familiar to a learner so that feedback can be tailored to the vocabulary or linguistic constructions they are familiar with. Finally, in security, it is highly relevant to identify the regional variant of the language used by a writer or poster.

Shared tasks on discriminating similar languages were organized in 2014 (Zampieri et al., 2014) and 2015 (Zampieri et al., 2015). This year's task continues in the same track, removing some of the easy languages (Czech and Slovak; Macedonian and Bulgarian), providing additional material for some of the harder variants (Serbo-Croat-Bosnian; Indonesian and Malay; Portuguese; Spanish), and adding new groups or variants (Mexican Spanish; French from Canada and France).

Like previous years, we relied on character ngram features, and a mixture of discriminative and generative statistical classifiers. Due to lack of time, we decided to eschew a full optimization of the feature sets and model combination, despite the fact that it provided excellent results in previous years (Goutte et al., 2014; Malmasi and Dras, 2015). Instead, we focused on two issues: the influence of the amount of data on the performance (open versus closed data), and the difference between a two-stage approach (predicting language/group, then variant) and a flat approach predicting the variant directly. To be clear, the "Advances" in the title of this paper do not relate to the performance and model we used this year, which are mostly similar to successful models of prior years. The intent is to advance our *understanding* of how these models works and what configurations are more effective.

An overview of the results of this shared task is presented in the shared task report (Malmasi et al., 2016). It provides a wider context for interpreting the results reported here, which we only compare to a few top systems. The shared task report also provides references to related work. The reader may also

Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects,
pages 178–184, Osaka, Japan, December 12 2016.

lang	DSLCC v1.0	DSLCC v2.1	DSL 2016	crawl	Total (open)
bs	20,000	20,000	20,000	-	60,000
hr	20,000	20,000	20,000	-	60,000
sr	20,000	20,000	20,000	-	60,000
es-AR	20,000	20,000	20,000	-	60,000
es-ES	20,000	20,000	20,000	-	60,000
es-MX	-	20,000	20,000*	-	40,000*
fr-CA	-	-	20,000	40,000	60,000
fr-FR	-	-	20,000	-	20,000
id	20,000	20,000	20,000	-	60,000
my	20,000	20,000	20,000	-	60,000
pt-BR	20,000	20,000	20,000	-	60,000
pt-PT	20,000	20,000	20,000	-	60,000

Table 1: Statistics on the training data used for training our systems. (*: 2016 data was actually identical to DSLCC v2.1 data)

find a lot of references to related work (within and outside the shared tasks) in Section 2 of (Goutte et al., 2016).

In the following section, we describe the data we worked with, the features we extracted from the data, and the models we trained on these features. Section 3 summarizes our results on the shared task test set and compared them to a few key systems from other participants. Finally, we discuss those results and their significance in Section 4.

2 Data and Methods

We now describe the data we used for our two runs, the features we extracted from the data, and the models we trained on those features.

2.1 Data

In order to evaluate the impact of using additional data on the performance of the discriminative performance, we built "closed" systems on the 2016 training data only, and "open" systems using additional data.

The closed systems use the data provided for the 2016 evaluation only. This consisted of 20k sentences from the news domain for each of the twelve language variant, for a total of 240k sentences (column **DSL 2016** in Table 1). We joined the `train` and `dev` portions of the training data as we evaluate the performance using ten fold cross-validation rather than using a single development set.

The open systems used data from previous DSL shared tasks (DSLCC v1.0 and DSLCC v2.1, (Tan et al., 2014)), plus additional text crawled from the web site of the Québec journal *La Presse*. For each of the variants used in previous years, this results in 60k sentences per variant (20k per corpus). Mexican Spanish was not included in previous years, but the DSLCC v2.1 corpus released last year contained 20k sentences for that variant, for use in the "unshared" task that (unfortunately) received little attention. We did not realize before training our system that the 20k sentences provided for `es-MX` this year were actually identical to the material provided last year for that variant, which means that our material for the `es-MX` variant is actually the same 20k sentences duplicated. For `fr-CA`, we added 40k sentences from the web crawl of *La Presse*. We checked that the added material did not overlap with the material provided by the DSL organizers (for training or testing). For French, our training material was therefore unbalanced, with only 20k sentences for `fr-FR` versus 60k for `fr-CA`.

Due to lack of time, we did not take part in the Arabic dialect sub-task, despite its great interest.

2.2 Features

Character ngram counts have been popular and effective features since at least (Cavnar and Trenkle, 1994), and produced top results at previous evaluations (Goutte et al., 2014; Malmasi and Dras, 2015; Goutte and Leger, 2015). We therefore relied again on character ngrams. This year, however, we only used 6grams. The reasons for this choice are multiple:

- Optimizing the size and combination of ngram features produces small performance improvements. However this optimization also requires significant effort and time, which we did not have this year.
- In our experience from previous shared tasks, 6grams were almost always the best feature set. When they were not, they were very close.
- Our main focus this year was not on maximizing performance of a single system, but on investigating the influence of training data size and the difference between a flat and two-stage model.

Using character 6grams therefore ensures that we build systems with reasonable (if not top) performance, while removing the variability in the choice (or optimization) of the features. It allows us to evaluate the impact of the data size and model architecture, everything else being kept equal. In addition, as we had a limited number of hours to spend on the shared task, we avoided the effort associated with feature engineering and optimization.

2.3 Models

Having decided on the feature set and training data, we tried two competing approaches.

2.3.1 `run1`

In the 2014 and 2015 evaluations (Goutte et al., 2014; Goutte and Leger, 2015) , we used a two stage approach, where we

1. train a level-1 classifier to predict the language group, and
2. train a level-2 classifier to predict the variant within each group.

This approach is actually not too costly. On the one hand, the lower level classifiers are often binary classifiers trained on smaller amounts of data. On the other hand, the top level classifier focuses on a simpler task, with bigger differences between groups, so a simple multiclass probabilistic classifier can be trained efficiently with almost perfect performance.

Our `run1` implements this two-stage approach again. The top-level classifier is a probabilistic classifier (Gaussier et al., 2002) similar to Naïve Bayes, trained in a single pass over the data, making it suitable for large-scale training. Using ten fold cross-validation, we estimate that the error rate of that first stage on the open task is below 0.051% (335 errors out of 660k examples), i.e. roughly one mistake per 2000 examples.

The level-2 classifiers are Support Vector Machines (SVM) trained using SVM^{light} (Joachims, 1998). For the three groups with only two variants (French, Portuguese and Indonesian/Malay), we trained a single binary SVM classifier in each group, taking one variant as the positive class, and the other variant as the negative class. For the two groups with three variants (Spanish and Serbo-Croat-Bosnian), we trained three SVM classifiers in one-versus-all configuration: each variant is taken as the positive class, in turn, with the other two as the negative class. The outputs of the three classifiers are calibrated to make them comparable (Bennett, 2003). At prediction time, we pick the class with the highest calibrated classifier output. Although training SVM models can be slow for large datasets, we restrict ourselves to linear kernels, and we only use the examples within a group to estimate the model, which is a maximum of 180k examples. Once the SVM classifiers and calibration layers have been estimated, prediction is just a few dot-products away, and therefore extremely fast. In previous years, we also validated that this combination of discriminative approach and calibration provides slightly better performance than modeling the problem directly with a multiclass probabilistic classifier.

Run	A	B1	B2	Overall	(CV)
run1	**89.03**	**94.80**	**90.00**	**89.29**	92.50
run2	88.77	94.20	89.00	88.98	92.63
SUKI	88.37	82.20	79.60	87.79	
Citius	87.10	66.40	69.20	85.62	

Table 2: Predictive accuracy (in %) for the 2016 open track runs, for our two runs and two runner-ups, on the three official test sets and overall. Rightmost column gives cross-validation estimate, for comparison.

2.3.2 run2

In the 2015 evaluation (Zampieri et al., 2015), the best performing system in the closed task (Malmasi and Dras, 2015) used a "flat" approach, treating the entire problem as a single, multiclass classification, with excellent results, only slightly below the best overall performance on the open task for test set A and best overall for test set B.

We attempt to test this flat approach on our chosen feature set, as our run2. Note that the best approach in (Malmasi and Dras, 2015) uses an *ensemble* of classifiers trained on different feature spaces (words, word bigrams, character bigrams, 4grams and 6grams). As we focus on a single feature set, we did not reproduce the ensemble part of that approach. The key difference between run1 and run2 is really the two-stage vs. flat approach.

We use again Support Vector Machines trained using SVM^{light} in one-versus-all fashion. For each of the 12 variants, a binary classifier is trained using one variant as the positive class and the rest as negative examples. The output of each of the 12 classifiers is then calibrated into a proper probability (Bennett, 2003). At prediction time, a sentence is sent through each calibrated classifier, producing a proper probability. The prediction is the class with highest probability. Note that despite its conceptual simplicity, this approach is more costly than the two-stage approach, as it requires training 12 binary classifier on 660k examples each (for the *open* track; 240k for the *closed* track). In addition, class imbalance is more severe for this model.

3 Results

We made four submissions for each of the three test sets (A, B1 and B2): two models on the *open* and two models on the *closed* tracks. The performance of each model was also estimated on the full training set (train+dev partitions of the official data) using a stratified ten fold cross-validation.

When the test data was received, we simply processed test set A as it was provided, as it seemed to match the training data fairly well. For the twitter data (test sets B1 and B2), we did light preprocessing by removing markers for accounts and hashtags (@ and #), as well as URLs. For example the tweet:

```
RT @xxmarioo: savage #KimExposedTaylorParty https://t.co/7FpfbmqziQ
```

is turned, before being sent to the classifiers, into:

```
RT xxmarioo: savage KimExposedTaylorParty
```

Official results on the three test sets were obtained from the organizers. From these results, we compute an overall score which is the micro-averaged accuracy over classes and test sets. The two 'B' test sets contain only 500 exemples vs. 12,000 for test set 'A', so the overall score is a weighted average of the provided accuracies, with eights 12/13, 1/26 and 1/26 for A, B1 and B2, respectively. Note that, in a single-label multiclass evaluation like this one, micro-averaged accuracy, precision, recall and (as a consequence) F-scores are identical. They differ slightly from the *weighted* F_1 used as the official ranking metric, but differences are small, probably due to the fact that classes are balanced in the test data.

Table 2 shows the results for the *open* track, while Table 3 shows the results for the *closed* track. As a naïve baseline, we propose to use a "group-perfect random" baseline, i.e. a classifier that would correctly identify the language group (a very easy task) and would perform randomly on the variants

Run	A	B1	B2	Overall	(CV)
run1	88.59	91.40	**87.80**	88.67	89.26
run2	88.12	90.80	86.60	88.16	88.87
tubasfs	**89.38**	86.20	82.20	**88.98**	
GWU	88.70	**92.00**	**87.80**	88.79	

Table 3: Predictive accuracy (in %) for the 2016 closed track runs, on the three official test sets and overall. Rightmost column gives cross-validation estimate, for comparison.

	sr-hr-bs	español	français	id-my	portugês
sr-hr-bs	2994	3	1	1	1
español	1	2990	3	1	5
français	0	0	1996	0	4
id-my	1	0	2	1997	0
portugês	0	1	0	0	1999

Table 4: Language group confusion on test set A, run1: reference in rows, predicted in columns.

within a group. The accuracy of this baseline is $\frac{\#\text{groups}}{\#\text{variants}}$ which is 41.67% for test set A and 40% for test sets B1 and B2, resulting in an overall score of 41.54%.

According to the official ranking,[1] our run1 results yield the top performance in the *open* track, closely followed by our run2 results and the two runner-up systems (SUKI and Citius_Ixa_Imaxin). Another participant submitted only for the twitter data (B1 and B2) and is not included in Table 2.

On the closed track, our results are slightly below the top two systems overall (tubasfs and GWU), with slight variations across the three test sets. Our run1 yield top results on test set B2 and close on test set B1 (the difference amounts to 3 tweets out of 500), but was outperfomed on the larger test set A. Note that tubasfs, GWU and our run1 are within 0.3% of each other, which may not be highly significant, either practically or statistically. A more precise assessment of the significance of the differences ill require access to the individual predictions.

Table 4 shows the confusion table between language groups for run1 on test set A (other runs and conditions are similar). Overall, there are 16 to 24 language group mistakes on test set A, depending on the track, i.e. below 0.2% error rate. Although still very low, this is quite significantly above the cross-validation estimate of 0.05%. The reasons for this will require further investigation. Most mistakes are, as expected, between Spanish, Portuguese and/or French, but a few are surprising (e.g., two Indonesian sentences predicted as French). On test sets B, the only mistake observed on either run or condition is a Portuguese user predicted as Bosnian. Overall this suggests that the first stage group classifier has little impact on performance, as it costs us at most 0.2% in error rate.

Looking at the language variant confusion in Table 5 shows that errors are not uniformly distributed. There are more confusions between Serbian and Croatian in the news data, and about as many confusions between Bosnian and Serbian in the twitter data. The confusion between Croatian and Bosnian is consistently smaller than for the other two pairs. In Spanish, errors appear unbalanced, with many Mexican sentences incorrectly assigned to the other two variants. This is likely due to a combination of the smaller size of the Mexican data, and the fact that we duplicated the data by mistake, which underestimates the unbalance between the classes.

4 Discussion

In light of these results and considering the questions we were targetting, we can reach the following conclusions.

- Data size has a small but consistent impact on performance. Keeping the models equal, the difference in performance brought by training on the *open* data was 0.72% on average. As this involves

[1] Available from http://ttg.uni-saarland.de/vardial2016/dsl2016.html.

set:	A			B1+B2		
	sr	hr	bs	sr	hr	bs
sr	692	198	109	179	8	13
hr	112	880	6	13	195	2
bs	85	24	888	8	0	192

set:	A			
es-	-ar	-es	-mx	
-ar	945	32	19	
-es	81	878	35	
-mx	175	152	673	

set:	A	
fr-	-ca	-fr
-ca	937	63
-fr	77	919

set:	A	
	id	my
id	990	7
my	14	986

set:	A		B1+B2	
pt-	-br	-pt	-br	-pt
-br	956	44	189	11
-pt	59	940	21	179

Table 5: Language variant confusion for run1: reference in rows, predicted in columns.

training on three times more data, whether this is worth it in practice is debatable, but it clearly brought our run1 above the best *closed* data result.

- The two-stage approach of predicting the group performs slightly but consistently better than the "flat" approach of predicting the variant directly. Keeping the data equal, the difference in performance between run1 and run2 was 0.41%, on average, in favour of the former. Again, this may not be a significant difference in practice, but given the advantage of the two stage approach in terms of training time, we think that this provides a convincing argument in favour of that approach. A side conclusion is that this suggests that the gain observed last year in the winning system (Malmasi and Dras, 2015) may be due to the ensemble combination, which could also be applied to the two stage approach.

- Our systems performed rather well on the twitter data, which seemed to be a challenge for several participants. Although that data was expected to be of lower quality than the journalistic material (language variety, frequent code switching and inventive character sequences), we also had a lot more material: segments in test set A had up to 88 words, whereas segments in test sets B1 and B2 had up to 6400. This was clearly helpful by providing better ngram statistics. It also helped that English was not among the candidate languages/variants as a lot of tweet material is clearly English. It would be interesting to check performance on single tweets.

- Previous work on twitter suggested that removing hashtags and account names altogether may yield a small performance gain (Lui and Baldwin, 2014). In this work, we decided to remove the # and @ characters alone, with the motivation that the hashtag or account *text* itself may point to the correct variant. A systematic evaluation of the different strategies is left to future work, although based on results from Lui and Baldwin (2014), we conjecture that it is unlikely to make a significant difference.

- The cross-validation estimates computed on the joint train+dev data yield optimistic estimates, especially on the open data. Although differences are expected, it is unusually large, and may suggest a domain mismatch between the test data and the training material. Another factor is that classes in the training data were imbalanced (fewer fr-FR and es-MX examples), whereas the test set is balanced. As a consequence, the fr-MX class is underpredicted compared to other Spanish variants. We did not observe the same effect on French, so this is still up for investigation.

- Our experience this year suggests that focusing on 6grams and removing the system combination (or ensemble) step makes it possible to set up competitive systems in very short time. The top performance this year was 89.3% accuracy, which is lower than last year, but still competitive (and on different test sets).

Acknowledgements

We wish to thank Marc Tessier at NRC for helping us acquire additional Canadian French data; the organizers for their hard work and for extending the submission deadline; the (anonymous) reviewers for many excellent suggestions.

References

Paul N. Bennett. 2003. Using asymmetric distributions to improve text classifier probability estimates. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 111–118, New York, NY, USA. ACM.

William Cavnar and John Trenkle. 1994. N-gram-based text categorization. *3rd Symposium on Document Analysis and Information Retrieval (SDAIR-94)*.

Éric Gaussier, Cyril Goutte, Kris Popat, and Francine Chen. 2002. A hierarchical model for clustering and categorising documents. In *Proceedings of the 24th BCS-IRSG European Colloquium on IR Research: Advances in Information Retrieval*, pages 229–247, London, UK, UK. Springer-Verlag.

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 (LT4VarDial)*, pages 78–84, Hissar, Bulgaria.

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

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating Similar Languages: Evaluations and Explorations. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Thorsten Joachims. 1998. Text categorization with Suport Vector Machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer.

Marco Lui and Timothy Baldwin. 2014. Accurate language identification of twitter messages. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 17–25, Gothenburg, Sweden, April. Association for Computational Linguistics.

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)*, pages 35–43, Hissar, Bulgaria.

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.

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.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A report on the dsl shared task 2014. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects (VarDial)*, pages 58–67, Dublin, Ireland.

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)*, pages 1–9, Hissar, Bulgaria.