Discriminating between Similar Languages using Weighted Subword Features

Adrien Barbaresi
Austrian Academy of Sciences (ÖAW-AC), Vienna
Berlin-Brandenburg Academy of Sciences and Humanities (BBAW)
adrien.barbaresi@oeaw.ac.at

Abstract

The present contribution revolves around a contrastive subword n-gram model which has been tested in the *Discriminating between Similar Languages* shared task. I present and discuss the method used in this 14-way language identification task comprising varieties of 6 main language groups. It features the following characteristics: (1) the preprocessing and conversion of a collection of documents to sparse features; (2) weighted character n-gram profiles; (3) a multinomial Bayesian classifier. Meaningful bag-of-n-grams features can be used as a system in a straightforward way, my approach outperforms most of the systems used in the DSL shared task (3rd rank).

1 Introduction

Language identification is the task of predicting the language(s) that a given document is written in. It can be seen as a text categorization task in which documents are assigned to pre-existing categories. This research field has found renewed interest in the 1990s due to advances in statistical approaches, and it has been active ever since, particularly since the methods developed have also been deemed relevant for text categorization, native language identification, authorship attribution, text-based geolocation, and dialectal studies (Lui and Cook, 2013).

As of 2014 and the first Discriminating between Similar Languages (DSL) shared task (Zampieri et al., 2014), a unified dataset (Tan et al., 2014) comprising news texts of closely-related language varieties has been used to test and benchmark systems. The documents to be classified are quite short and may even be difficult to distinguish for human annotators, thus adding to the difficulty and the interest of the task. A second shared task took place in 2015 (Zampieri et al., 2015). An analysis of recent developments can be found in Goutte el al. (2016) as well as in the report on the third shared task (Malmasi et al., 2016).

The present study was conducted on the occasion of the fourth VarDial workshop (Zampieri et al., 2017). It focuses on submissions to the DSL task, a 14-way language identification task comprising varieties of six main language groups: Bosnian (bs), Croatian (hr), and Serbian (sr); Argentine (es-AR), Peruan (es-PE), and Peninsular Spanish (es-ES); Dari Persian (fa-AF) and Farsi/Iranian Persian (fa-IR); Québec French (fr-CA) and Hexagonal French (fr-FR); Malay (*Bahasa Melayu*, my) and Indonesian (*Bahasa Indonesia*, id); Brazilian Portuguese (pt-BR) and European Portuguese (pt-PT).

Not all varieties are to be considered equally since differences may stem from extra-linguistic factors. It is for instance assumed that Malay and Indonesian derive from a millenium-old *lingua franca*, so that shorter texts have been considered to be a problem for language identification (Bali, 2006). Besides, the Bosnian/Serbian language pair seems to be difficult to tell apart whereas Croatian distinguishes itself from the two other varieties mostly because of political motives (Ljubeši[Pleaseinsertintopreamble] et al., 2007; Tiedemann and Ljubešić, 2012).

The remainder of this paper is organized as follows: in section 2 the method is presented, it is then evaluated and discussed in section 3.

2 Method

2.1 Preprocessing

Preliminary tests have shown that adding a custom linguistic preprocessing step could slightly

Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 184–189,
Valencia, Spain, April 3, 2017.

improve the results. As such, instances are tokenized using the *SoMaJo* tokenizer (Proisl and Uhrig, 2016), which achieves state-of-the-art accuracies on both web and CMC data for German. As it is rule-based, it is deemed efficient enough for the languages of the shared task. No stop words are used since relevant cues are expected to be found automatically as explained below. Additionnally, the text is converted to lowercase as it led to better results during development phase on 2016 data.

2.2 Bag of n-grams approach

Statistical indicators such as character- and token-based language models have proven to be efficient on short text samples, especially character n-gram frequency profiles from length 1 to 5, whose interest is (*inter alia*) to perform indirect word stemming (Cavnar and Trenkle, 1994). In the context of the shared task, a simple approach using n-gram features and discriminative classification achieved competitive results (Purver, 2014). Although features relying on the output of instruments may yield useful information such as POS-features (Zampieri et al., 2013), the diversity of the languages to classify as well as the prevalence of statistical methods call for low-resource methods that can be trained and applied easily.

In view of this I document work on a refined version of the *Bayesline* (Tan et al., 2014) which has been referenced in the last shared task (Barbaresi, 2016a) and which has now been used in official competition. After looking for linguistically relevant subword methods to overcome data sparsity (Barbaresi, 2016b), it became clear that taking frequency effects into consideration is paramount. As a consequence, the present method grounds on a bag-of-n-grams approach. It first proceeds by constructing a dictionary representation which is used to map words to indices. After turning the language samples into numerical feature vectors (a process also known as vectorization), the documents can be treated as a sparse matrix (one row per document, one column per n-gram).

Higher-order n-grams mentioned in the development tests below use feature hashing, also known as the "hashing trick" (Weinberger et al., 2009), where words are directly mapped to indices with a hashing function, thus sparing memory. The upper bound on the number of features has been fixed to 2^{24} in the experiments below.

2.3 Term-weighting

The next step resides in counting and normalizing, which implies to weight with diminishing importance tokens that occur in the majority of samples. The concept of term-weighting originates from the field of information retrieval (Luhn, 1957; Sparck Jones, 1972). The whole operation is performed using existing implementations by the *scikit-learn* toolkit (Pedregosa et al., 2011), which features an adapted version of the *tf-idf* (term-frequency/inverse document-frequency) term-weighting formula.[1] Smooth *idf* weights are obtained by systematically adding one to document frequencies, as if an extra document was seen containing every term in the collection exactly once, which prevents zero divisions.

2.4 Naive Bayes classifier

The classifier used entails a conditional probability model where events represent the occurrence of an n-gram in a single document. In this context, a multinomial Bayesian classifier assigns a probability to each target language during test phase. It has been shown that Naive Bayes classifiers were not only to be used as baselines for text classification tasks. They can compete with state-of-the-art classification algorithms such as support vector machines, especially when using approriate preprocessing concerning the distribution of event frequencies (Rennie et al., 2003); additionally they are robust enough for the task at hand, as their decisions may be correct even if their probability estimates are inaccurate (Rish, 2001).

2.5 "Bayesline" formula

The *Bayesline* formula used in the shared task grounds on existing code (Tan et al., 2014)[2] and takes advantage of a comparable feature extraction technique and of a similar Bayesian classifier. The improvements described here concern the preprocessing phase, the vector representation, and the parameters of classification. Character n-grams from length 2 to 7 are taken into account.[3]

[1] http://scikit-learn.org/stable/modules/feature_extraction.html

[2] https://github.com/alvations/bayesline

[3] `TfidfVectorizer(analyzer='char', ngram_range=(2,7), strip_accents=None, lowercase=True)` followed by `MultinomialNB(alpha=0.005)`, adapted from https://web.archive.org/web/20161215142013/http://scikit-learn.org/stable/auto_examples/text/document_classification_20newsgroups.html

N-gram length	2	3	4	5	6	7	8*	9*
1	.690	.794	.852	.882	.894	**.902**	.895	.895
2	.705	.798	.854	.883	.895	**.902**	.899	.899
3		.808	.859	.884	.896	**.902**	.901	.901

Table 1: Benchmark by F1-weighted of a common range of n-gram length combinations on 2016 DSL data (*=hashed features)

3 Evaluation

3.1 Data from the third edition

In order to justify the choice of the formula, experiments have been conducted on data from the third edition of the DSL shared task (Malmasi et al., 2016); training and development sets have been combined as training data, and gold data used for evaluation. The method described above has been tested with several n-gram ranges; the results are summarized in Table 1. The best combinations were found with a minimum n-gram length of 1 to 3 and a maximum n-gram length of 6 to 8. Accordingly, an *aurea mediocritas* from 2 to 7 has been chosen.

Table 2 shows the extraction, training, and testing times for n-gram lengths with a mininum of 2. One can conclude that the method is computationnally efficient on the shared task data. Execution with feature hashing is necessary for higher-order n-grams due to memory constraints; it effectively improves scalability but it also seems to be a trade-off between computational efficiency and accuracy, probably due to the upper bound on used features and/or hash collisions.

Range	Extraction	Training	Testing
2,2	19	0.3	0.0
2,3	41	1.0	0.0
2,4	72	2.0	0.1
2,5	136	4.4	0.3
2,6	230	8.6	0.5
2,7	387	14.0	0.9
2,8*	179	15.4	0.9
2,9*	208	18.2	1.1

Table 2: Evolution of execution time (in seconds) with respect to n-gram length (*=hashed features)

Table 3 documents the efficiency and accuracy of several algorithms on the classification task, without extensive parameter selection. The Ridge (Rifkin and Lippert, 2007) and Naive Bayes classifiers would have outperformed the best submission of the 2016 competition (0.894) with scores of respectively 0.895 and 0.902, while the Passive-Aggressive (Crammer et al., 2006) and Linear Support Vector (Fan et al., 2008) classifiers would have been ranked second with a score of 0.892. It is noteworthy that the Naive Bayes classifier would still have performed best without taking the development data into consideration (accuracy of 0.898).

3.2 Data from the fourth edition

As expected, the method performed well on the fourth shared task, as it reached the 3rd place out of 11 teams (with an accuracy of 0.925 and a weighted F1 of 0.925). In terms of statistical significance, it was ranked first (among others) by the organizers. The official baseline/Bayesline used a comparable algorithm with lower results (accuracy and weighted F1 of 0.889).

The confusion matrix in Figure 1 details the results. Three-way classifications between the variants of Spanish and within the Bosnian-Croatian-Serbian complex still leave room for improvement, although Peruvian Spanish does not seem to be as noisy as the Mexican Spanish data from the last edition. The F-score on variants of Persian is fairly high (0.960) which proves that the method can be applied to a wide range of alphabets.

The same method has been tested without preprocessing on new data consisting in the identification of Swiss German dialects (GDI shared task). The low result (second to last with an accuracy of 0.627 and a weighted F1 of 0.606) can be explained by the lack of adaptation, most notably to the presence of much shorter instances. The classification of the Lucerne variant is particularly problematic, it calls for tailored solutions.

4 Conclusion

The present contribution revolves around a contrastive subword n-gram model which has been tested in the *Discriminating between Similar Languages* shared task. It features the following char-

Type	Training (s)	Accuracy	F1-weighted
Naive Bayes	14	.902	.902
Bernoulli NB	16	.882	.883
Nearest Centroid/Rocchio	33	.759	.760
Stochastic Gradient Descent	464	.813	.813
Perceptron	764	.884	.884
Passive-Aggressive	947	.892	.892
Linear Support Vector Classifier	1269	.892	.892
Ridge Classifier	1364	.895	.895

Table 3: Comparison of several classifier types on the extracted feature vectors, ordered by ascending training time (in seconds) on data from 2016. Classifiers used without extensive parameter tuning, linear SVC and SGD with L2 penalty.

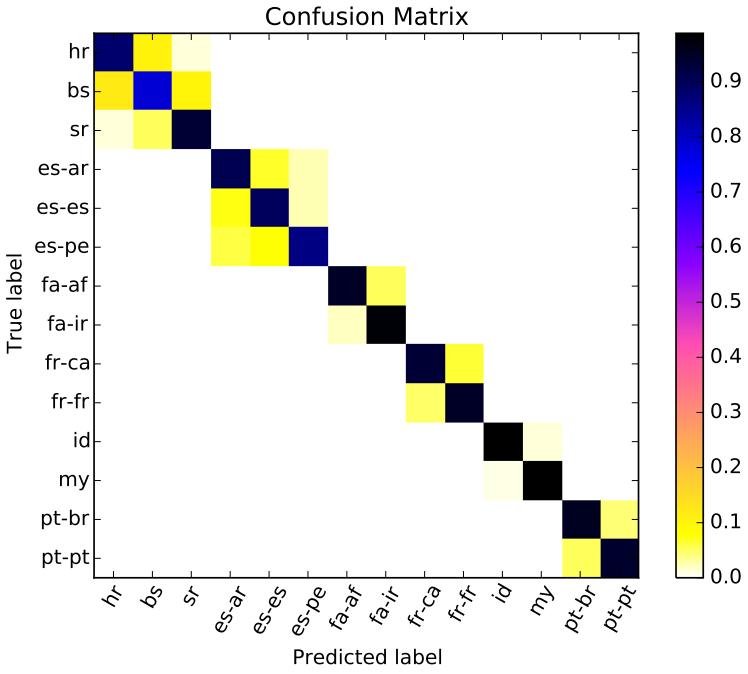

Figure 1: Confusion matrix for DSL task (closed training, 2017 data)

acteristics: (1) the conversion of a collection of preprocessed documents to a matrix of sparse *tf-idf* features; (2) weighted character n-gram profiles; (3) a multinomial Bayesian classifier, hence the name "Bayesline". Meaningful bag-of-n-grams features can be used as a system in a straightforward way. In fact my method outperforms most of the systems used in the DSL shared task.

Thus, I propose a new baseline and make the necessary components available under an open source licence.[4] The *Bayesline* efficiency as well as the difficulty to reach higher scores in open training could be explained by artificial regularities in the test data. For instance, the results for the Dari/Iranian Persian and Malay/Indonesian pairs are striking, these clear distinctions do not reflect the known commonalities between these language varieties. This could be an artifact of the data, which feature standard language of a different nature than the continuum "on the field", that is between two countries as well as within a single country. The conflict between in-vitro and real-world language identification has already been emphasized in the past (Baldwin and Lui, 2010); it calls for the inclusion of web texts (Barbaresi, 2016c) into the existing task reference.

[4]https://github.com/adbar/vardial-experiments

Acknowledgments

Thanks to the anonymous reviewers for their comments.

References

Timothy Baldwin and Marco Lui. 2010. Language Identification: The Long and the Short of the Matter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 229–237. Association for Computational Linguistics.

Ranaivo-Malançon Bali. 2006. Automatic Identification of Close Languages–Case Study: Malay and Indonesian. *ECTI Transaction on Computer and Information Technology*, 2(2):126–133.

Adrien Barbaresi. 2016a. An Unsupervised Morphological Criterion for Discriminating Similar Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 212–220. The COLING 2016 Organizing Committee.

Adrien Barbaresi. 2016b. Bootstrapped OCR error detection for a less-resourced language variant. In Stefanie Dipper, Friedrich Neubarth, and Heike Zinsmeister, editors, *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, pages 21–26. University of Bochum.

Adrien Barbaresi. 2016c. Efficient construction of metadata-enhanced web corpora. In *Proceedings of the 10th Web as Corpus Workshop*, pages 7–16. Association for Computational Linguistics.

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*, pages 161–175.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, 7:551–585.

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.

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)*, pages 1800–1807. European Language Resources Association (ELRA).

Nikola Ljubešić, Nives Mikelić, and Damir Boras. 2007. Language identification: how to distinguish similar languages? In *29th International Conference on Information Technology Interfaces*, pages 541–546. IEEE.

Hans Peter Luhn. 1957. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of research and development*, 1(4):309–317.

Marco Lui and Paul Cook. 2013. Classifying English Documents by National Dialect. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 5–15.

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)*.

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.

Thomas Proisl and Peter Uhrig. 2016. SoMaJo: State-of-the-art tokenization for German web and social media texts. In *Proceedings of the 10th Web as Corpus Workshop*, pages 57–62. Association for Computational Linguistics.

Matthew Purver. 2014. A Simple Baseline for Discriminating Similar Languages. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 155–160.

Jason D. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. 2003. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 616–623. ACM.

Ryan M. Rifkin and Ross A. Lippert. 2007. Notes on Regularized Least Squares. Technical report, MIT-CSAIL.

Irina Rish. 2001. An Empirical Study of the Naive Bayes Classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, pages 41–46. IBM New York.

Karen Sparck Jones. 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28(1):11–21.

Liling Tan, Marcos Zampieri, Nikola Ljubešić, 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*, pages 11–15.

Jörg Tiedemann and Nikola Ljubešić. 2012. Efficient discrimination between closely related languages. In *Proceedings of COLING*, pages 2619–2633.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature Hashing for Large Scale Multitask Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM.

Marcos Zampieri, Binyam Gebrekidan Gebre, and Sascha Diwersy. 2013. N-gram language models and POS distribution for the identification of Spanish varieties. In *Proceedings of TALN 2013*, pages 580–587.

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*, 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*, 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 Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*.