Computationally efficient discrimination between language varieties with large feature vectors and regularized classifiers

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

Abstract

The present contribution revolves around efficient approaches to language classification which have been field-tested in the *Vardial* evaluation campaign. The methods used in several language identification tasks comprising different language types are presented and their results are discussed, giving insights on real-world application of regularization, linear classifiers and corresponding linguistic features. The use of a specially adapted Ridge classifier proved useful in 2 tasks out of 3. The overall approach (XAC) has slightly outperformed most of the other systems on the DFS task (Dutch and Flemish) and on the ILI task (Indo-Aryan languages), while its comparative performance was poorer in on the GDI task (Swiss German dialects).

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, a unified dataset (Tan et al., 2014) comprising news texts of closely-related language varieties has been used to test and benchmark systems. The instances 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. An analysis of recent developments can be found in Goutte el al. (2016), in the reports on previous shared tasks as well as in a recently published survey on language and dialect identification (Jauhiainen et al., 2018). In previous editions the shared tasks organized at VarDial included dialects of Arabic, German, and the DSL shared task which featured similar languages and language varieties (Zampieri et al., 2017; Malmasi et al., 2016; Zampieri et al., 2015; Zampieri et al., 2014). Other related shared tasks are the MGB challenge on Arabic (Ali et al., 2017), the PAN lab on author profiling which included dialects and language varieties (Rangel et al., 2017), and the TweetLID shared task which included similar languages (Zubiaga et al., 2016).

The present study was conducted on the occasion of the fifth VarDial workshop (Zampieri et al., 2018). It focuses on submissions to three different datasets: the first iteration of the Discriminating between Dutch and Flemish in Subtitles (DFS) task (van der Lee and van den Bosch, 2017), the second iteration of work on Swiss German dialects based on the GDI dataset as described in Samardžić et al. (2016), and the first iteration of the Indo-Aryan Language Identification (ILI) shared task, based on the compiled ILI dataset (Kumar et al., 2018). The peculiarities of the datasets include their diversity in terms of linguistic characteristics and their fluctuating difficulty, as most varieties can be mutually intelligible with diverging degrees of lexical and morphosyntactic variation. As in previous tasks, the number of instances is limited in size, the training and test sets are on the order of magnitude of thousands or tens

Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 164–171
Santa Fe, New Mexico, USA, August 20, 2018.

of thousands of instances, the latter being fairly small in size, with at most a sentence each time. As a consequence, a classifier constructed on small training sets may be biased and have a large variance, as the classifier parameters (coefficients) are poorly estimated. It is also unstable, as small changes in the training set may cause large changes in the classifier. A key component of winning systems is their capacity to make correct predictions on unseen data based on the trained model.

In this context, it has been shown that more conventional statistical methods can very well be more accurate than latest machine learning approaches (Barbaresi, 2017), resulting in a paradox common to a fair number of applications: "Knowing that a certain sophisticated method is not as accurate as a much simpler one is upsetting from a scientific point of view as the former requires a great deal of academic expertise and ample computer time to be applied." (Makridakis et al., 2018)

The remainder of this paper is organized as follows: in section 2 the preprocessing and feature extraction steps are presented, the classifiers follow in section 3, and three systems are then evaluated and discussed in section 4.

2 Large feature vectors

2.1 Preprocessing

Preliminary tests have shown that adding a custom linguistic preprocessing step could slightly 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 supposed to be 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. Additionally, the text is converted to lowercase (if applicable) as it led to better results during tests on training data, mostly because of the potential noise induced by words at the beginning of a sentence.

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 annotation tools may yield useful information such as POS-features (Zampieri et al., 2013), the varieties to classify here are less-resourced in terms of tools, which calls 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 and used in previous editions (Barbaresi, 2016a; Barbaresi, 2017). After looking for linguistically relevant subword methods to overcome data sparsity, 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).

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; Spärck 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. 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. In addition, the feature vectors have been normalized using L2-norm, which led to marginal improvements.

Overall, the feature vectors are typically large and lead to high-dimensional sparse datasets, so that computationally efficient methods are called for. An additional constraint resides in finding classifiers which work well with sparse matrices.

3 Classifiers

3.1 Naive Bayes classifier

All naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. The classifier used entails a conditional probability model where events represent the occurrence of a n-gram in a single document. In this context, a multinomial Bayesian classifier assigns a probability to each target language during test phase, as categorical and multinomial distributions are conflated. This approach performs well with comparatively small training data, as the estimate of the parameters necessary for classification is good enough to compete with more complex approaches. In the case of large-scale data, it is computationally very efficient as it allows for classification in near linear time, i.e. reading the training data once and then reading the instances of the test data.

It has been shown that Naive Bayes classifiers were not merely 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).

3.2 "Bayesline" formula

The *Bayesline* formula used in the shared task grounds on existing code (Tan et al., 2014) and takes advantage of a comparable feature extraction technique and of a similar Bayesian classifier. This approach outperformed most systems in the previous edition of the shared task (Barbaresi, 2017). It has been refined for this year's edition concerning the vector representation and the parameters of classification. After cross-validation tests on the training data parameters have been added compared to the default procedure, most importantly a regularization parameter as described above and another parameter for additive smoothing, also known as Laplace or Lidstone smoothing, which has been set to 0.04 instead of 0.005. Additive smoothing allows the assignment of non-zero probabilities to features which have not been seen during training. Character n-grams from varying lengths are taken into account and then the classification takes place.[1]

One of the potential shortcomings of this approach is that it does not see the instance space as a high dimensional space, but just as a collection of frequencies from which it estimates the probability of each class using the Bayes theorem.

3.3 Regularized Linear Classifiers

Regression analysis consists in estimating the relationship between a dependant variable (here the target language) and a number of predictors (here the linguistic cues). During training, a function of the independent variables is estimated, the fitted model is then used on test data. Generalized Linear Models consist in a regression in which the target value is expected to be a linear combination of the input variables. They can for example be used to discriminate between web texts for inclusion into web corpora for linguistic research (Barbaresi, 2015).

Ridge is a regularization technique also known as Ridge regression or Tikhonov regularization (Tikhonov, 1943; Hoerl, 1962). It is particularly useful when the number of input variables greatly exceeds the number of observations and when there are many small to medium-sized effects, which is often the case for n-gram data. In such cases, the least square regression estimator may not uniquely exist, and although it uses a biased estimator Ridge regression can reduce the expected squared loss. This

[1] *TfidfVectorizer(analyzer='char', ngram_range=(2,6), strip_accents=None, lowercase=True, sublinear_tf=True, smooth_idf=False, use_idf=True, min_df=0, norm='l2')* followed by *MultinomialNB(alpha=0.04)*, adapted from https://web.archive.org/web/20180507114732/http://scikit-learn.org/stable/auto_examples/text/document_classification_20-newsgroups.html See also https://github.com/adbar/vardial-experiments

Classifier	F1 (macro)
Random Baseline	0.5000
Naive Bayes	0.6207
SGD	0.6134
Ridge	**0.6318**

Table 1: Results for DFS task, all classifiers used the same data preparation pipeline

Classifier	F1 (macro)
Random Baseline	0.2521
Naive Bayes	**0.6336**
SGD	0.6291
Ridge	0.6296

Table 2: Results for GDI task, all classifiers used the same data preparation pipeline

method also proves useful in the presence of multicollinearity, that is when the predictor variables used in a regression are highly correlated, which can be expected at least in some cases for natural language. In practice, it imposes a penalty on the size of coefficients so that they become more robust to collinearity. Moreover the ridge parameter allows a linear regression to work in cases that are not completely linearly separable. From an practical point of view, ridge regression can avoid overfit through regularization, as it shrinks the coefficients towards zero (but not exactly zero) to reduce variance. This step can minimize the impact of statistically irrelevant features on the trained model, in this regard it is bound to simplify the model. In the present case, it is expected to focus on salient features and lead to a faster, more clear-cut classification. However, ridge regression cannot perform variable selection directly, thus the potential increase in prediction accuracy cannot be immediately used for interpretation.

The method used for the task is close to linear least-squares Support Vector Machine (SVM) classification in terms of formulation but faster in practice. The implementation in the Python framework *scikit-learn* (Pedregosa et al., 2011) includes a stochastic average gradient descent solver which is known to work well with large-scale and sparse machine learning problems often encountered in text classification and natural language processing.

To provide a basis for comparison, further experiments have been conducted using a stochastic gradient descent (SGD) classifier, a method which aims at finding minima or maxima by iteration and is thus more computationally complex but still an efficient approach to fit linear models, useful when the number of samples (and the number of features) is very large. Consequently, SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing. In the implementation used here, the regularized linear model with stochastic gradient descent (SGD) learning is equivalent to a (soft-margin) linear Support Vector Machine (SVM), thus providing a comparison with a well-known classification technique. As SGD requires a number of hyperparameters, parameter tuning using grid search has been performed on the training data using k-fold cross-validation tests.

4 Evaluation

The results for the DFS task are summarized in Table 4. The chosen n-gram window was 2 to 6 characters, the feature extraction and classification processes have been conducted as described above. The classification methods used yield relatively low improvements with respect to the *Bayesline*, which shows this is a challenging task. The multinomial Naive Bayes (run 1) indeed outperformed the SGD classifier (run 2) although the latter had been optimized during training by using parameter tuning. The Ridge classifier was significantly more robust during training (higher cross-validation scores) and effectively reached a slightly higher score in the test run. This submission has been ranked 3rd out of 7. The confusion matrix shown in Figure 1 depicts a rather balanced mix of errors.

The same method has been tested with the same data preprocessing in the identification of Swiss

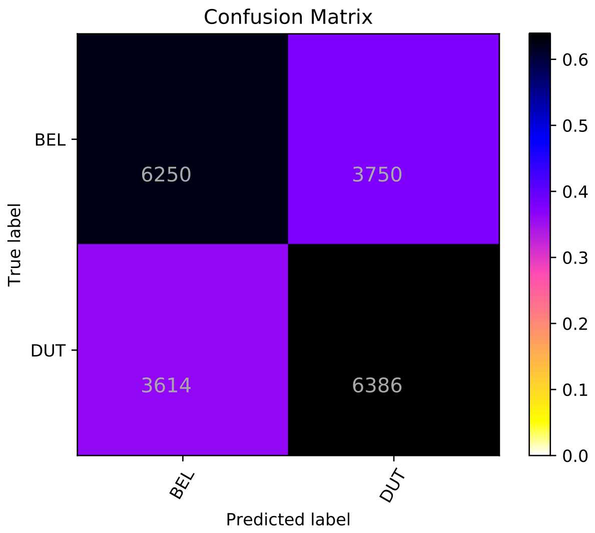

Figure 1: Confusion matrix for the Ridge classifier on the DFS task

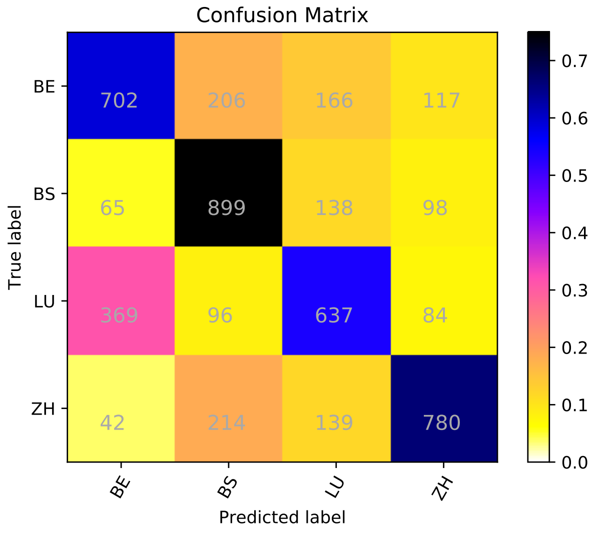

Figure 2: Confusion matrix for the Naive Bayes classifier on the GDI task

German varieties (GDI shared task), using only the training data from this year's edition. Contrarily to the other tasks, a larger N-Gram span has been selected (1 to 6) as this seemed to carry a little more useful information. The classifiers have been optimized during training in a similar way. The results are summarized in Table 2.

This task has seen the worst result compared to the other teams, with the best submission ranked 3rd out of 4. This result is in line with the fact that the Naive Bayes classifier outperformed more complex methods, which shows they could not gain more fine-grained information or generalize better on the test data. As in the last edition of the shared task, the classification of the Lucerne variant is more problematic than the other ones as shown in Figure 2, however the Ridge classifier is more robust so that the gap is closing. The F1 score is significantly higher than in the previous competition – 0.634 against 0.606 (Barbaresi, 2017). The improvements can be explained by potentially cleaner data, focus on F1 instead of accuracy during model selection and parameter tuning, whereas the comparatively low score can be explained by the nature of the occurrences to discriminate: they are much shorter and less regular than in the other tasks, orthographic normalization is a potential problem and the instances seem to be closer to speech corpora. Additionally, they had already been tokenized, so that the tokenization during

Classifier	F1 (macro)
Random Baseline	0.2024
Naive Bayes	0.8540
SGD	0.8833
Ridge	**0.8983**

Table 3: Results for the ILI task, all classifiers used the same data preparation pipeline

pre-processing is without effect here.

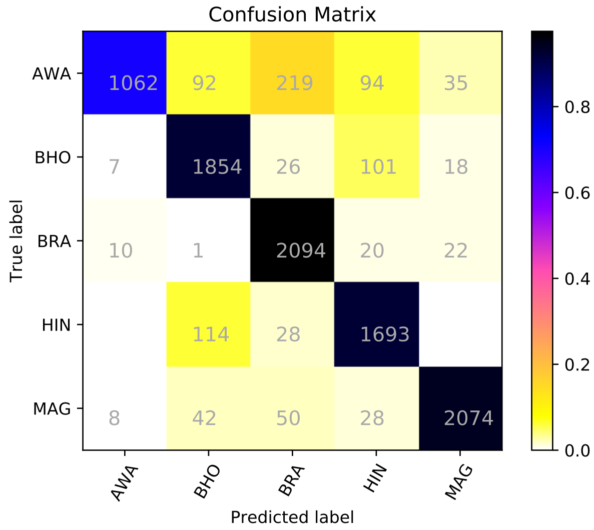

Figure 3: Confusion matrix for the Ridge classifier on the ILI task

Last, the Indo-Aryan varieties have seen the Naive Bayes classifier lead to a significantly lower score than the other methods, which highlights their comparatively good performance and also explains why the best submission (Ridge classifier) has been ranked 2nd out of 6 in the competition, which demonstrates empirically that the processing chain described here also works well for other alphabets. The results are summarized in Table 3, the finer discrimination using regularized classifiers can be explained by the morphological characteristics of Indo-Aryan languages and the subsequent necessity to better assess the statistical significance of the extracted n-grams. The confusion matrix in Figure 3 highlights the difficulty of the AWA variety, which has the worst recall whereas the BRA variety seems to be easier to discriminate from the rest.

5 Conclusion

The present contribution deals with computationally efficient discrimination between language varieties using large feature vectors and regularized classifiers. The methods described have been tested in the *VarDial* evaluation campaign. The characteristics of the shared tasks (most notably the limited training data and the relatively short length of instances) call for specially adapted solutions. Supervised optimization during the training phase shows that there is a major proportion of small to medium-sized effects, whereas the high-frequency spectrum could even be ignored without significantly impacting performance. This situation implies that it may be more difficult to avoid overfitting with more powerful methods, so that statistical models such as linear classifiers are not only computationally efficient but also lead to better results in practice because of their less precise modelization of phenomena. Moreover, as information in the low frequency spectrum is valuable, not performing feature selection and using regularization can often allow for better predictive performance. In a comparison of discriminative and generative learning as typified by logistic regression and naive Bayes, it has been shown that a generative classifier may also approach its (higher) asymptotic error much faster (Ng and Jordan, 2002) which partly explains why the classifier works better as training data is limited in size.

The *Bayesline* efficiency as well as the difficulty to reach higher scores in open training could be explained by these characteristics and also by artificial regularities in the test data. 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, 2016b) into the existing task reference.

Future work includes further refinements of classification methods. Reducing the dimensionality of datasets (for example by principal component analysis) can pave the way for more complex classifiers, however no performance improvement seems within easy reach so far. Another more promising option with respect to previous shared tasks could consist of bagging linear models, which may be a way to produce finer estimates without causing the models to overfit.

References

Ahmed Ali, Stephan Vogel, and Steve Renals. 2017. Speech Recognition Challenge in the Wild: Arabic MGB-3. *arXiv preprint arXiv:1709.07276.*

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.

Adrien Barbaresi. 2015. *Ad hoc and general-purpose corpus construction from web sources.* Ph.D. thesis, École Normale Supérieure de Lyon.

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, Osaka, Japan. The COLING 2016 Organizing Committee.

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

Adrien Barbaresi. 2017. Discriminating between similar languages using weighted subword features. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 184–189, Valencia, Spain, April.

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.

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).

Arthur E. Hoerl. 1962. Application of Ridge analysis to regression problems. *Chemical Engineering Progress*, 58:54–59.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2018. Automatic language identification in texts: A survey. *arXiv preprint arXiv:1804.08186.*

Ritesh Kumar, Bornini Lahiri, Deepak Alok, Atul Kr. Ojha, Mayank Jain, Abdul Basit, and Yogesh Dawar. 2018. Automatic Identification of Closely-related Indian Languages: Resources and Experiments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC).*

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.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2018. Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889.

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.

Andrew Y. Ng and Michael I. Jordan. 2002. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press.

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.

Francisco Rangel, Paolo Rosso, Martin Potthast, and Benno Stein. 2017. Overview of the 5th author profiling task at PAN 2017: Gender and language variety identification in Twitter. *Working Notes Papers of the CLEF*.

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.

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.

Tanja Samardžić, Yves Scherrer, and Elvira Glaser. 2016. ArchiMob–A corpus of spoken Swiss German. In *Proceedings of the Language Resources and Evaluation (LREC)*, pages 4061–4066, Portoroz, Slovenia).

Karen Spärck 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.

Andrey Nikolayevich Tikhonov. 1943. On the stability of inverse problems. *Doklady Akademii Nauk SSSR*, 39(5):195–198.

Chris van der Lee and Antal van den Bosch. 2017. Exploring Lexical and Syntactic Features for Language Variety Identification. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 190–199, Valencia, Spain.

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 (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.

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)*, Valencia, Spain.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Santa Fe, USA.

Arkaitz Zubiaga, Inaki San Vicente, Pablo Gamallo, José Ramom Pichel, Inaki Alegria, Nora Aranberri, Aitzol Ezeiza, and Víctor Fresno. 2016. TweetLID: a benchmark for tweet language identification. *Language Resources and Evaluation*, 50(4):729–766.