When Sparse Traditional Models Outperform Dense Neural Networks: the Curious Case of Discriminating between Similar Languages

Maria Medvedeva[♡] **Martin Kroon**[♢] **Barbara Plank**[♢]

[♡]Dept. of Computational Linguistics, Saarland University, Saarbrücken, Germany

[♢]Center for Language and Cognition Groningen, University of Groningen, The Netherlands

mariam@coli.uni-saarland.de,martinkroon06@gmail.com,b.plank@rug.nl

Abstract

We present the results of our participation in the VarDial 4 shared task on discriminating closely related languages. Our submission includes simple traditional models using linear support vector machines (SVMs) and a neural network (NN). The main idea was to leverage language group information. We did so with a two-layer approach in the traditional model and a multi-task objective in the neural network. Our results confirm earlier findings: simple traditional models outperform neural networks consistently for this task, at least given the amount of systems we could examine in the available time. Our two-layer linear SVM ranked 2nd in the shared task.

1 Introduction

The problem of automatic language identification has been a popular task for at least the last 25 years. From early on, different solutions showed very high results (Cavnar et al., 1994; Dunning, 1994), while the more recent models achieve near-perfect accuracies.

Distinguishing closely-related languages, however, still remains a challenge. The *Discriminating between similar languages* (DSL) shared task (Zampieri et al., 2017) is aimed at solving this problem. For this year's task our team (mm_lct) built a model that discriminates between 14 languages or language varieties across 6 language groups (which had two or three languages or language varieties in them).[1]

The most popular of the more recent systems, such as `langid.py` (Lui and Baldwin, 2012) and CLD/CLD2[2] produce very good results based on datasets containing fewer than 100 languages, but even a model trained on as many as 131 languages (Kocmi and Bojar, 2017) and whatlang (Brown, 2013) with trained on 184 and 1100 languages, are not able to distinguish closely-related (and therefore very similar) languages and dialects to a satisfying degree, at least not to the extent of the data available.

As part of the DSL 2017 shared task we chose to further explore traditional linear approaches, as well as deep learning methods. In the next Section we shortly discuss previous approaches to the task of discriminating between similar languages. Then in Section 3 we describe our systems and the data, followed by the results in Section 4, which are discussed in Section 5. We conclude in Section 6.

2 Related Work

Even though a number of researches in dialect identification have been conducted, (Tiedemann and Ljubešić, 2012; Lui and Cook, 2013; Maier and Gómez-Rodriguez, 2014; Ljubešić and Kranjcic, 2015, among many others), they mostly deal with particular language groups or language variations. We saw as our goal to create a language identifier that is able to produce comparable results for languages within all provided groups with the same set of features for every language group, so that it can be expanded outside those languages provided by the DSL shared task without any changes other than to the training corpus – as to make the system as language-independent and universal as possible.

Most of the language identifiers that use linear classifiers rely on character n-gram models (Carter et al., 2011; Ng and Selamat, 2011; Zampieri and Gebre, 2012) and combinations of character and word n-grams (Milne et al., 2012;

[1] The term *language* shall henceforth be used for both 'language' and 'language variety'.

[2] `https://github.com/CLD2Owners/cld2`

Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 156–163,
Valencia, Spain, April 3, 2017. ©2017 Association for Computational Linguistics

Vogel and Tresner-Kirsch, 2012; Goldszmidt et al., 2013), also including top systems from previous DSL shared tasks (Goutte and Léger, 2015; Malmasi and Dras, 2015; Çöltekin and Rama, 2016).

The overviews of the previous DSL shared tasks (Zampieri et al., 2014; Zampieri et al., 2015; Goutte et al., 2016) showed that SVMs always produce some of the top results in this task, especially when tested on same-domain datasets (Çöltekin and Rama, 2016). Thus, we chose to put our efforts into improving upon SVM approaches, but still decided to experiment with an neural network to see if we could get comparable results, while using fewer features and reducing the chance of overfitting.

The popularity of using NNs for NLP tasks is growing. A few neural language identifiers already exist as well (Tian and Suontausta, 2003; Takçi and Ekinci, 2012; Simões et al., 2014, among others), however on average traditional systems still seem to outperform them. The results of the DSL 2016 shared task also show the same tendency overall (Bjerva, 2016; Cianflone and Kosseim, 2016; Çöltekin and Rama, 2016; Malmasi et al., 2016).

3 Methodology and Data

In this section, we first describe the datasets that were provided for the DSL 2017 shared task. Then we describe the three systems we used to tackle the problem: first a two-layer SVM that uses language-group classification, then a single-layer SVM that does not use grouping and finally an neural network-based approach.

3.1 Data

This year's data is a new version of the DSL *Corpus Collection* (DSLCC) (Tan et al., 2014), with again 18,000 instances for training and 2,000 instances for development. The test data consists of 1,000 instances per language and contains the same languages as the training and development data. The test data is furthermore very similar to the development data, as supported by the results – during-development performance was almost the same as the performance on the test set. All instances come from short newspaper texts.

However whereas last year's version of the DSLCC contained Mexican Spanish, this year's version has Peruvian Spanish (`es-PE`). Another new addition is the Farsi language group, with the two variations Persian (`fa-IR`) and Dari (`fa-AF`). Thus, this year's version contains 14 languages belonging to 6 groups:

- BCS: containing Bosnian, Croatian and Serbian;
- Spanish: containing Argentine, Peninsular and Peruvian varieties;
- Farsi: containing Afghan Farsi (or Dari) and Iranian Farsi (or Persian);
- French: containing Canadian and Hexagonal varieties;
- Indonesian and Malay; and
- Portuguese: containing Brazilian and European varieties.

An overview of the data is given in Table 1, which includes the number of instances as well as the number of tokens for each language in the training and development data.

In the final submissions we performed no preprocessing on the data. During development we explored the usefulness of replacing all characters for lower case, having placeholders for numbers and removing punctuation, but we found that it decreased performance of the system.

Finally, for the final submission we have had all our runs trained on the combination of both training and development datasets, as has been shown to be effective by last year's winning team (Çöltekin and Rama, 2016).

3.2 Run 1 – SVM with grouping

As our first, most promising run we have developed and submitted a two-layer classifier, which first predicts for all instances which language group it belongs to, and then classifies the specific languages within the guessed language groups. This method has been used by DSL participants before (Franco-Salvador et al., 2015; Nisioi et al., 2016), and has shown to have a positive impact on the performance. Adopting this method, we have built a combination of SVMs with linear kernels.

The first SVM is for deciding on the language group to which the language belongs. As features it uses character-based uni- to 6-grams (including whitespace and punctuation characters) weighted

		Training		Dev.	
Language	Code	Instances	Tokens	Instances	Tokens
Croatian	hr	18,000	658,492	2,000	72,731
Bosnian	bs	18,000	555,680	2,000	61,574
Serbian	sr	18,000	606,403	2,000	66,494
Argentine Spanish	es-AR	18,000	746,531	2,000	83,090
Peninsular Spanish	es-ES	18,000	789,870	2,000	88,116
Peruvian Spanish	es-PE	18,000	455,630	2,000	51,021
Dari	fa-AF	18,000	501,157	2,000	55,249
Persian	fa-IR	18,000	659,040	2,000	72,894
Canadian French	fr-CA	18,000	510,134	2,000	55,934
Hexagonal French	fr-FR	18,000	746,531	2,000	68,136
Indonesian	id	18,000	595,187	2,000	64,749
Malay	my	18,000	453,326	2,000	50,692
Brazilian Portuguese	pt-BR	18,000	695,826	2,000	76,694
European Portuguese	pt-PT	18,000	638,124	2,000	71,153

Table 1: The number of instances and number of tokens for all languages in the training data and the development data.

by tf-idf.[3] While testing it on the development set it appeared to be very reliable, as all misclassified instances on the group level contained only names and digits and were, therefore, impossible to be classified by a human either.

The second SVM predicts the specific languages within each group (with the same feature parameters for every group), using word-based uni- and bigrams, in combination with character-based n-grams up to 6 characters weighted by tf-idf, as well.

Figure 1a shows that when trained on a subset of 100,000 randomly selected instances (while keeping the language distribution the same) of the training data, the best accuracy is achieved when using character n-grams from 1 to 6 characters and no word n-grams. However, when we trained and tested it on the DSL 2016 data, it scored lower than the winning team (for the in-domain test set). We therefore chose a different set of features by adding word unigrams and bigrams that gave us a slight advantage over last year's task's results. It did, though, reduce the performance on this year's development, but the reduction was so minimal that we deemed it unlikely to be significant (accuracies of 0.90296 without word n-grams vs. 0.90206 with word uni- and bigrams), especially when considering that the difference between the accuracies becomes smaller the more training data is available.

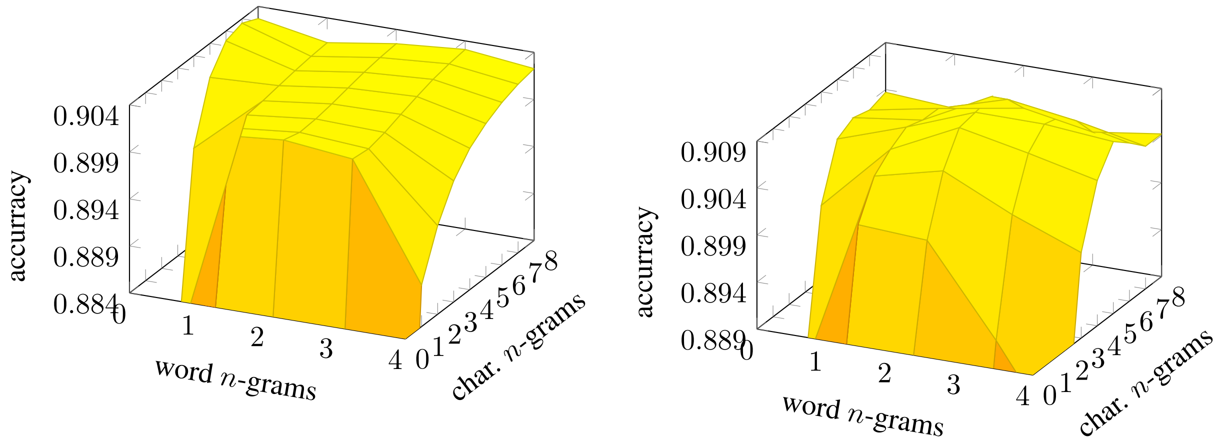
Fine-tuning the second SVM for particular language groups seemed to defeat the goal of developing a language-independent classifier – retraining on other languages would have not been possible, without largely adjusting the system.

3.3 Run 2 – SVM without grouping

As the second run we submitted a single system, a linear kernel SVM that does not use language-group classification first but classifies languages straight away. When exploring different combinations of word and character n-grams we trained the system on the 100,000 same instances and found that the highest results were achieved with a combination of word uni- and bigrams and character uni- to 6-grams (see Figure 1b). Thus, for this run we have the same parameters as the *within-groups* classifier of run 1.

When trained on this year's full training set and tested on the development set, this system performs slightly better than the two-layer system

[3] The formula used to compute tf-idf is as follows, as defined by scikit-learn Python package: $\text{tf-idf}(d,t) = \text{tf}(t) * \text{idf}(d,t)$ where $\text{idf}(d,t) = \log(n/\text{df}(d,t)) + 1$ where n is the total number of documents and $\text{df}(d,t)$ is the document frequency; the document frequency is the number of documents d that contain term t (Pedregosa et al., 2011).



(a)

(b)

Figure 1: Visualisation of the differences in accuracy with changing maximum lengths of word and character n-grams trained on 100,000 instances of training data and tested on the development dataset. Where n-grams are 0, n-grams were turned off; the left lower corner, therefore, is the random baseline. (a) shows the accuracies for the SVM with grouping, (b) for the SVM without grouping.

(but likely to be insignificantly better, with a less than 0.1% difference in accuracy).

3.4 Run 3 – CBOW multi-task NN

We also experimented with NNs, in particular, an NN with a multi-task objective. The idea was to take advantage of language group information to guide learning. This represents a complimentary approach to run 1.

Our preliminary experiments confirmed earlier findings that NN-based approaches are outperformed by more simple linear models for language identification (Çöltekin and Rama, 2016; Gamallo et al., 2016). We compared recurrent NNs to simpler models based on continuous bag of word (CBOW) representations (Mikolov et al., 2013), which are similar to feedforward NNs and simply take the mean vector of the input embeddings as input representation. CBOW was not only quicker to train, it also outperformed their RNN/LSTM counterparts, thus resulting in our final submission.

In particular, run 3 is a simple CBOW NN with two output layers: the first predicting the actual language identifier, the second predicting the language group. The CBOW multi-task NN training objective is to minimise the cross-entropy loss on language identity (L_1) and language group identification (L_2), weighted by λ set on the development set and trained on a subset of 10,000 instances. The joined training objective was:

$$L = (1 - \lambda)L_1 + \lambda L_2, \quad \text{where} \quad \lambda = 0.1$$

As input features it uses embeddings on character uni- to 5-grams, which outperforms simple word input alone. We observed that the multi-task objective sped up learning, although ultimately the difference between an MTL and a non-MTL counterpart was minor. We submitted the MTL model as final run. It was trained on the joined training and development data without any preprocessing, as to make it more comparable to our SVM submissions.

Note that due to time constraints we did not fully explore many directions here, like feature space, hyperparameters or alternative models, but overall NN seemed less promising for this task.

4 Results

Based on absolute scores, our first system (SVM with grouping) performed second best in the DSL shared task (Zampieri et al., 2017) with an accuracy of 0.9254. Both our other systems also performed substantially higher than the random baseline of 0.0714: accuracies of 0.9236 and 0.8997 for the SVM without grouping and the NN, respectively. See Table 2 for an overview of the accuracies and F$_1$-scores of our three systems.

Table 3 presents the confusion matrix for the SVM with grouping. Out-of-group confusions – which are very rare in general, in all three runs –

Run	**Accuracy**	**F_1 (micro)**	**F_1 (macro)**	**F_1 (weighted)**
Random baseline	0.0714			
SVM with grouping	0.9254	0.9254	0.9250	0.9250
SVM without grouping	0.9226	0.9226	0.9221	0.9221
CBOW NN	0.8997	0.8997	0.9001	0.9001

Table 2: Accuracies and F_1-scores (micro, macro and weighted) for the three systems, along with the random baseline.

	hr	bs	sr	es-AR	es-ES	es-PE	fa-AF	fa-IR	fr-CA	fr-FR	id	my	pt-BR	pt-PT
hr	894	92	13	0	0	0	0	0	0	1	0	0	0	0
bs	120	760	119	0	1	0	0	0	0	0	0	0	0	0
sr	11	71	918	0	0	0	0	0	0	0	0	0	0	0
es-AR	0	0	0	846	69	80	0	0	0	1	0	0	3	1
es-ES	0	0	0	62	893	42	0	0	0	1	0	0	1	1
es-PE	0	0	0	20	29	951	0	0	0	0	0	0	0	0
fa-AF	0	0	0	0	0	0	968	32	0	0	0	0	0	0
fa-IR	0	0	0	0	0	0	27	972	0	1	0	0	0	0
fr-CA	0	0	0	0	0	0	0	0	948	52	0	0	0	0
fr-FR	1	0	1	0	0	0	0	0	61	937	0	0	0	0
id	0	0	0	0	2	0	0	0	0	1	987	10	0	0
my	0	0	0	0	0	0	0	0	0	2	14	984	0	0
pt-BR	0	0	0	0	0	0	0	0	0	2	0	0	943	55
pt-PT	0	0	0	1	1	0	0	0	0	1	0	0	43	954

Table 3: Confusion matrix for the SVM with grouping.

occur notably less often with the SVM with grouping (only 2.2% of the confusions it makes are out-of-group confusions) than with the other runs. This is to be expected as the SVM with grouping is designed to group instances of the same language group together and then to discriminating between the particular language variations within the groups. Within-group confusions also occur relatively less often with the SVM with grouping (in all groups, except for French, the accuracy is higher for the SVM with grouping than the SVM without grouping; the NN has notably lower accuracies for all groups: see Table 4).

	SVM-1	**SVM-2**	**NN**
hr-bs-sr	0.8579	0.8518	0.8295
es	0.8991	0.8986	0.8657
fa	0.9705	0.9680	0.9505
fr	0.9434	0.9449	0.9340
id-my	0.9880	0.9840	0.9659
pt	0.9509	0.9498	0.9256

Table 4: Accuracies for all language groups for the first SVM (with grouping), the second SVM (without grouping) and the NN.

Overall, fewest within-group confusions occurred in the Indonesian-Malay group. The most mistakes were made in the BSC group. This is also supported by the accuracies. The values, though, do not necessarily support claims that Bosnian, Serbian and Croatian must then be more alike than, e.g., Indonesian and Malay are: differences in the amount of training data or the quality of the data may cause incomparable results. Also the language groups that contain three languages perform, as expected, overall worse than the groups with two languages.

Another striking aspect of the confusion matrix is that, in the BSC group, Bosnian seems to be confused more than Croatian or Serbian. Serbian and Croatian are rarely confused with each other. This suggests that in a gradual transition between Croatian and Serbian, Bosnian is somewhere in the middle. A similar gradual transition does not seem to exist for the Spanish varieties (as supported by the confusion matrix).

This is also supported by the fact that Bosnian, of all 14 languages, performs the worst in terms of both precision and recall ($F_1 = 0.79$). Indonesian and Malay both perform the best, both with an almost perfect $F_1 = 0.99$. A full report of language-specific performances for the SVM with grouping can be found in Table 5.

	Precision	**Recall**	**F$_1$-score**
hr	0.87	0.89	0.88
bs	0.82	0.76	0.79
sr	0.87	0.92	0.90
es-AR	0.91	0.85	0.88
es-ES	0.90	0.89	0.90
es-PE	0.89	0.95	0.92
fa-AF	0.97	0.97	0.97
fa-IR	0.97	0.97	0.97
fr-CA	0.94	0.95	0.94
fr-FR	0.94	0.94	0.94
id	0.99	0.99	0.99
my	0.99	0.98	0.99
pt-BR	0.95	0.94	0.95
pt-PT	0.94	0.95	0.95

Table 5: Language-specific performance measures for the SVM with grouping.

5 Discussion

We presented our approaches to tackling the problem of discriminating between similar languages and dialects. The SVM which first groups instances based on language group using word uni- and bigrams and character unigrams to 6-grams as features works best by a very small margin – in the DSL shared task it performed second in absolute F$_1$-scores, but also by a small margin.

The margin between our two SVMs, though, is so small that it might not even be statistically significant.[4] However, although grouping does not really improve the performance of the system, it does make the model noticeably faster. This is because, when grouping, the system requires less memory at once, as it fits the data for only one language group at a time, which is only about a sixth of the total data (in this dataset), depending on the group. It only processes the total amount of the data once – when grouping the instances in language groups, but then it uses fewer features.

As expected, the SVMs do perform notably better than the deep-learning approach we tried. However, our NN uses simple CBOW and still places itself rather well among other systems.

Figure 1a suggests that the two-layer SVM approach might perform slightly better when using no word n-grams altogether. Although we decided against such a system, it will be interesting to see what the impact of removing word n-grams for the two-layer SVM feature set will have on the performance of said approach. It would also be interesting to see if having only longer n-grams (i.e. only 3-5 character n-grams) or only combinations of particular lengths would improve the results.

6 Conclusions

Discriminating between similar languages is still not a fully solved problem – no known system reaches perfect performance. The models presented in this paper once again confirm that traditional models, such as SVMs, perform better on this task than deep learning techniques. We also showed that a two-layer approach in which languages are first classified based on language groups barely improves performance – yet, in our experience, it speeds up the system significantly.

References

Johannes Bjerva. 2016. Byte-based Language Identification with Deep Convolutional Networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 119–125, Osaka, Japan.

Ralf D. Brown. 2013. Selecting and weighting n-grams to identify 1100 languages. In *International Conference on Text, Speech and Dialogue*, pages 475–483. Springer.

[4] In fact, on the development dataset, the SVM with grouping performed slightly worse than the one that does not group – contrary to the performance on the test data.

Simon Carter, Manos Tsagkias, and Wouter Weerkamp. 2011. Semi-supervised priors for microblog language identification. In *Proceedings of the 11th Dutch-Belgian Information Retrieval Workshop (DIR 2011)*, pages 12–15.

William B. Cavnar, John M. Trenkle, et al. 1994. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175.

Çağri Çöltekin and Taraka Rama. 2016. Discriminating Similar Languages with Linear SVMs and Neural Networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 15–24, Osaka, Japan.

Andre Cianflone and Leila Kosseim. 2016. N-gram and Neural Language Models for Discriminating Similar Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 243–250, Osaka, Japan.

Ted Dunning. 1994. *Statistical identification of language*. Computing Research Laboratory, New Mexico State University.

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

Pablo Gamallo, Iñaki Alegria, José Ramom Pichel, and Manex Agirrezabal. 2016. Comparing Two Basic Methods for Discriminating Between Similar Languages and Varieties. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 170–177, Osaka, Japan.

Moises Goldszmidt, Marc Najork, and Stelios Paparizos. 2013. Boot-strapping language identifiers for short colloquial postings. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 95–111. Springer.

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

Tom Kocmi and Ondřej Bojar. 2017. Lanidenn: Multilingual language identification on character window. *arXiv preprint arXiv:1701.03338*.

Nikola Ljubešić and Denis Kranjcic. 2015. Discriminating between closely related languages on twitter. *Informatica*, 39(1):1.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30. Association for Computational Linguistics.

Marco Lui and Paul Cook. 2013. Classifying English documents by national dialect. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 5–15. Citeseer.

Wolfgang Maier and Carlos Gómez-Rodriguez. 2014. Language variety identification in Spanish tweets. In *Proceedings of the EMNLP2014 Workshop on Language Technology for Closely Related Languages and Language Variants*, 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 (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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Rachel Mary Milne, Richard A. O'Keefe, and Andrew Trotman. 2012. A study in language identification. In *Proceedings of the Seventeenth Australasian Document Computing Symposium*, pages 88–95. ACM.

Choon-Ching Ng and Ali Selamat. 2011. Improving language identification of web page using optimum profile. In *International Conference on Software Engineering and Computer Systems*, pages 157–166. Springer.

Sergiu Nisioi, Alina Maria Ciobanu, and Liviu P. Dinu. 2016. Vanilla Classifiers for Distinguishing between Similar Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 235–242, Osaka, Japan.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011.

Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Alberto Simões, José João Almeida, and Simon D. Byers. 2014. Language identification: a neural network approach. In *OASIcs-OpenAccess Series in Informatics*, volume 38. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Hidayet Takçi and Ekin Ekinci. 2012. Minimal feature set in language identification and finding suitable classification method with it. *Procedia Technology*, 1:444–448.

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 (BUCC)*, pages 11–15, Reykjavik, Iceland.

Jilei Tian and Janne Suontausta. 2003. Scalable neural network based language identification from written text. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–48. IEEE.

Jrg Tiedemann and Nikola Ljubešić. 2012. Efficient discrimination between closely related languages. In *Proceedings of COLING 2012*, pages 2619–2634, Mumbai, India, December. The COLING 2012 Organizing Committee.

John Vogel and David Tresner-Kirsch. 2012. Robust language identification in short, noisy texts: Improvements to liga. In *Proceedings of the Third International Workshop on Mining Ubiquitous and Social Environments (MUSE 2012)*, pages 43–50.

Marcos Zampieri and Binyam Gebrekidan Gebre. 2012. Automatic identification of language varieties: The case of Portuguese. In *KONVENS2012-The 11th Conference on Natural Language Processing*, pages 233–237. Österreichischen Gesellschaft für Artificial Intelligende (ÖGAI).

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.