Tübingen system in VarDial 2017 shared task: experiments with language identification and cross-lingual parsing

Çağrı Çöltekin
Department of Linguistics
University of Tübingen, Germany
ccoltekin
@sfs.uni-tuebingen.de

Taraka Rama
Department of Linguistics
University of Tübingen, Germany
taraka-rama.kasicheyanula
@uni-tuebingen.de

Abstract

This paper describes our systems and results on VarDial 2017 shared tasks. Besides three language/dialect discrimination tasks, we also participated in the cross-lingual dependency parsing (CLP) task using a simple methodology which we also briefly describe in this paper. For all the discrimination tasks, we used linear SVMs with character and word features. The system achieves competitive results among other systems in the shared task. We also report additional experiments with neural network models. The performance of neural network models was close but always below the corresponding SVM classifiers in the discrimination tasks.

For the cross-lingual parsing task, we experimented with an approach based on automatically translating the source treebank to the target language, and training a parser on the translated treebank. We used off-the-shelf tools for both translation and parsing. Despite achieving better-than-baseline results, our scores in CLP tasks were substantially lower than the scores of the other participants.

1 Introduction

In this paper, we describe our efforts in two rather different tasks during our participation in VarDial 2017 shared tasks (Zampieri et al., 2017). The first task, which we collectively call *language identification* task, aims to identify closely related languages or dialects. VarDial 2017 hosted three related language identification tasks: *Discriminating between similar languages* (DSL) shared task which includes closely related languages in six groups, *Arabic dialect identification* (ADI), and *German dialect identification* (GDI). The second task, *cross-lingual parsing* (CLP), aims to exploit resources available for a related source language for parsing a target language for which no syntactically annotated corpora (treebank) is available. This paper focuses on the language identification, while providing a brief summary of our methods and results for the CLP task as well.

Although language identification is a mostly solved problem, closely related languages and dialects still pose a challenge for the language identification systems (Tiedemann and Ljubešić, 2012; Zampieri et al., 2014; Zampieri et al., 2015; Zampieri et al., 2017). For this task, we experimented with two different families of models: linear support vector machines (SVM), and (deep) neural network models. For both models we used combination of character and word (n-gram) features. Similar to our earlier experiments in VarDial 2016 shared task (Çöltekin and Rama, 2016), the linear models performed better than the neural network models in all language identification tasks. We describe both families of models, and compare the results obtained. In the VarDial 2017 shared task campaign, the DSL and ADI shared tasks had both open and closed track submissions, while GDI had only closed tracks. For all the tasks, we only participate in the closed track.

While discriminating closely related languages is a challenge for the language identification task, the similarities can be useful in other tasks. By using information or resources available for a related (source) language one can build or improve natural language tools for a (target) language. This is particularly useful for low-resource languages, and tasks that require difficult-to-build language-specific tools or resources. Parsing fits into this category well, since treebanks, the primary resources used for parsing, require considerable time

Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 146–155,
Valencia, Spain, April 3, 2017. ©2017 Association for Computational Linguistics

and effort to create. Hence, transferring knowledge from one or more (not necessarily related) languages is studied extensively in some recent work and found to be useful (Yarowsky et al., 2001; Hwa et al., 2005; Zeman and Resnik, 2008; McDonald et al., 2011; Tiedemann et al., 2014a, just to name a few). Particularly, it has been shown that these approaches tend to perform better than purely unsupervised methods, which can be another natural choice for parsing a language without a treebank.

There are two common approaches for transfer parsing. The first one is often called *model transfer*, which typically involves training a delexicalized parser on the source language treebank, and using it on the target language, with further adaptation or lexicalization with the help of additional monolingual or parallel corpora (McDonald et al., 2011; Naseem et al., 2012). The second method is *annotation transfer*, which utilizes parallel resources to map the existing annotations for the source language to the target language (Yarowsky et al., 2001; Hwa et al., 2005; Tiedemann, 2014). In this work, we use a straightforward annotation-transfer method using freely available tools. Similar to the language identification, we only participated in the closed track of the CLP task.

The remainder of the paper is organized as follows. The next section provides brief descriptions of the tasks and the data sets. Section 3 describes the methods and the systems we used for both tasks, Section 4 presents our results and we conclude in Section 5 after a brief discussion.

2 Task description

In this section, we provide a brief description of the tasks, and the data sets. Detailed description of the task and data can be found in Zampieri et al. (2017).

2.1 Language identification

VarDial 2017 shared task included three language identification challenges.

- *Discriminating between similar languages* (DSL) shared task includes closely related languages in six groups:
  - Bosnian (bs), Croatian (hr) and Serbian (sr)
  - Malay (my) and Indonesian (id)
  - Persian (fa-ir) and Dari (fa-af)
  - Canadian (fr-ca) and Hexagonal French (fr-fr)
  - Brazilian (pt-br) and European Portuguese (pt-pt)
  - Argentine (es-ar), Peninsular (es-es), and Peruvian Spanish (es-pe)
- *Arabic dialect identification* task involves discriminating between five Arabic varieties:
  - Egyptian (egy)
  - Gulf (glf)
  - Levantine (lav)
  - North-African (nor)
  - Modern Standard Arabic (msa)
- *German dialect identification* (GDI) tasks involves identifying four Swiss German dialects from the following areas.
  - Basel (bs)
  - Bern (be)
  - Lucerne (lu)
  - Zurich (zh)

variety	characters		tokens	
	mean	sd	mean	sd
bs	196.53	90.80	30.86	14.18
hr	236.91	102.32	36.56	15.59
sr	209.13	97.47	33.64	15.45
es-ar	253.61	96.73	41.48	15.75
es-es	262.58	94.16	43.90	15.62
es-pe	148.48	79.66	25.33	13.26
fa-af	139.24	60.34	27.83	12.12
fa-ir	187.30	72.42	36.61	14.35
fr-ca	174.37	53.82	28.30	8.40
fr-fr	207.95	98.67	33.76	15.82
id	236.53	93.61	33.00	13.03
my	180.28	69.49	25.20	9.72
pt-br	235.51	96.82	38.63	15.66
pt-pt	217.59	90.21	35.46	14.58

Table 1: Average characters and space-separated tokens in the DSL data (training and development set combined).

The organizers provided separate training and development sets for the DSL task. The training set consists of 18 000 documents and the development set consists of 2 000 documents for each

variety	characters		tokens		docs
	mean	sd	mean	sd	
egy	141.50	200.63	25.74	35.78	3 415
glf	125.47	237.55	22.66	42.50	3 008
lav	105.48	145.35	19.37	26.03	3 308
msa	191.67	203.67	33.17	34.91	2 488
nor	80.30	121.13	14.41	21.06	3 305

Table 2: Average characters and space-separated tokens in the ADI data (training and development set combined).

variety	characters		tokens		docs
	mean	sd	mean	sd	
be	36.74	19.40	7.34	3.99	3 889
bs	44.75	26.38	8.41	4.97	3 411
lu	45.55	23.66	8.91	4.65	3 214
zh	39.11	21.57	7.24	3.96	3 964

Table 3: Average characters and space-separated tokens in the GDI data (only training set, no development set was porovided).

language variety. Although the data is balanced with respect to the number of documents, there is a slight variation with respect to the number of characters and tokens among different language varieties as presented in Table 1. These differences may explain some of the biases towards certain varieties within groups. Further details about the task and the data can be found in Goutte et al. (2016).

The ADI data includes transcriptions of speech from five different Arabic varieties. Besides the transcribed words, the ADI data also includes i-vectors, fixed-length vectors representing some acoustic properties of whole utterances. The ADI data shows slightly more class imbalance than the DSL data, as shown in Table 2. The lengths of the documents in the ADI data is also more varied. More information on the data and the task can be found in Malmasi et al. (2015).

The GDI task includes data from four Swiss German dialects. This data set includes much shorter documents compared to the DSL and ADI data sets. The GDI data statistics are also presented in Table 3.

2.2 Cross-lingual parsing

The cross lingual parsing tasks involved using one or more source language treebanks along with parallel texts to parse the target languages. The source–target language pairs for this task are,

- Target language: Croatian, Source language: Slovenian
- Target language: Slovak, Source language: Czech
- Target language: Norwegian, Source languages: Danish and Swedish

The source language treebanks are part of the Universal Dependencies (UD) version 1.4 (Nivre et al., 2016). The parallel texts are subtitles from the OPUS corpora collection (Tiedemann, 2012).

3 System descriptions

3.1 Language identification with SVMs

Similar to our past year's participation, we submitted results using a multi-class (one-vs-one) support vector machine (SVM) model. Unlike our last year's submissions (Çöltekin and Rama, 2016) where we used only character n-grams as features, we used a combination of both character and word n-grams. Both character and word n-gram features are weighted using sub-linear tf-idf scaling (Jurafsky and Martin, 2009, p.805). We did not apply any filtering (e.g., case normalization), except for removing features that occur in only a single document.

The ADI data set also included fixed-length numeric features, i-vectors, for each document. We concatenated these vectors with the tf-idf features in our best performing model for the ADI task. In all SVM models we combine the features in a flat manner and predict the varieties directly without using a two-stage or hierarchical approach. We also tuned the number of character and word n-grams, as well as the SVM margin parameter 'C' for each task separately. The SVMs were not very sensitive to the changes in these parameters. Table 4 lists the configurations of the SVM models in our main submission. We present further results on the effects of these parameters in Section 4. In all of our experiments, we combined the development and training sets for the DSL and ADI tasks and used 10-fold cross validation for tuning. We also used 10-fold cross validation for tuning the parameters of the system for the GDI task for which no designated development data was provided.

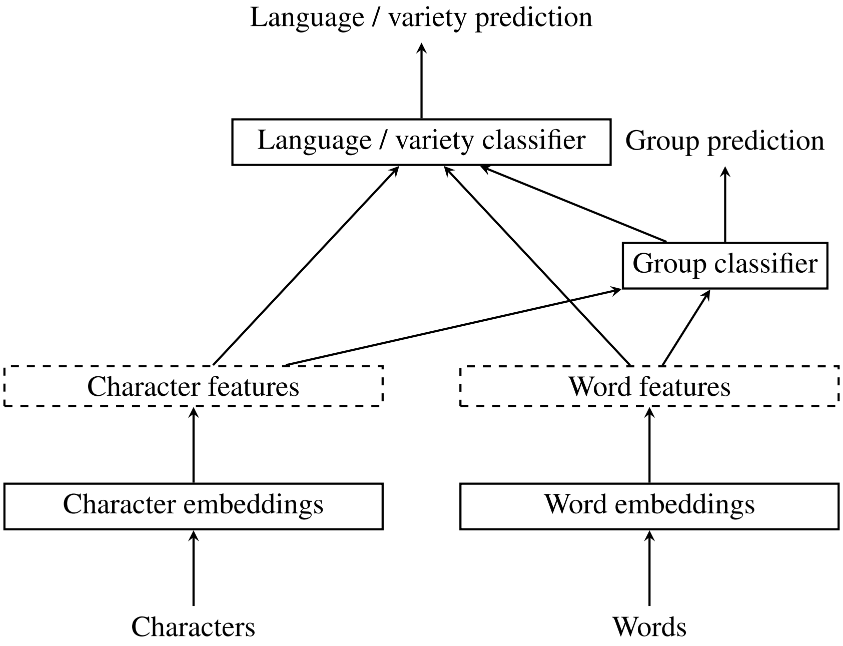

Figure 1: The schematic representation of our neural network architecture.

Task	word	char	C
DSL	3	7	1.8
ADI	3	10	0.5
GDI	2	7	0.7

Table 4: Maximum word and character n-grams, and the SVM margin parameter, C, used for each language identification task, for our main submission. We use all n-grams starting unigrams up to the indicated maximum n-gram value.

We also experimented with logistic regression, using both one-vs-rest and one-vs-one multi-class strategies. Like the previous year, the SVM models always performed slightly better than logistic regression models. In this paper, we only describe the SVM models and discuss the results obtained using them.

All linear models were implemented with scikit-learn (Pedregosa et al., 2011) and trained and tested using Liblinear backend (Fan et al., 2008).

3.2 Language identification with neural networks

The general architecture used for our hierarchical network model is presented in Figure 1. This is virtually identical to the general architecture described in Çöltekin and Rama (2016).

In this study, we use both task-specific character and word embeddings to train our model. They are trained during learning to discriminate the languages varieties. As opposed to general-purpose embeddings, they are expected to capture the input features (characters words) that are indicative of a particular language variety rather than words that are semantically similar.

The presented architecture is an instance of multi-label classification. During training, model parameters are optimized to guess both the group and the specific language variety correctly. Furthermore, we feed the model's prediction of the group to the classifier predicting the specific language variety. For instance, we would use the information that *fr-fr* and *fr-ca* labels belong to the French group. The intuition behind this model is that it will use the highly accurate group prediction during test time to tune into features that are useful within a particular language group for predicting individual varieties. For ADI, and GDI tasks, we do not use the group prediction since these data set contain only as single language group.

In principle, the boxes 'Group classifier' and 'Language / variety classifier' in Figure 1 may include multiple layers for allowing the classifier to generalize based on non-linear combinations in its input features. However, in the experiments reported in this paper, we did not use multiple layers in both the classifiers, since, it did not improve the results.

The dashed boxes in Figure 1 turn the sequence of word and character embeddings into fixed-size feature vectors. Any network layer/model that extracts useful features from a sequence of embeddings are useful here. The convolutional and recurrent neural networks are typical choices for this step. We have experimented with both methods, as well as simple averaging of embeddings.

In the experiments reported below, the documents are padded or truncated to 512 characters for the character embedding input, and they are padded or truncated to 128 tokens for the word embeddings input. For both embedding layers, we used dropout with rate 0.40. Both classifiers in the figure were single layer networks (with softmax activation function), predicting one-hot representations of groups and varieties. The network was trained using categorical cross-entropy loss function for both outputs using Adam optimization algorithm. To prevent overfitting, the training was stopped when validation set accuracy stopped improving after two iterations. All neural network experiments are realized using Keras (Chollet, 2015) with Tensorflow backend (Abadi et al., 2015).

3.3 Cross-lingual parsing

We adopted the *word-based MT* approach of Tiedemann et al. (2014b) for translating the source language dependency treebank(s) to target languages. In the first step, we used the *efmaral* system (Östling and Tiedemann, 2016) to word-align the OPUS parallel corpus of a source-target language pair. We word-aligned the parallel corpus from both source to target and target to source; and, then proceeded to symmetrize the alignments using *grow-diag-final-and* method. Then, we supplied the symmetric alignments to Moses (Koehn et al., 2007) and constrained the Moses system to train using phrase translations of length 1. Finally, we used the Moses decoder with the default settings to translate the source language treebank to target language. The intuition behind this approach is that word based translations do not require heuristics to correct the trees that result from the default phrase-based translation settings of Moses. We used this approach to create treebanks for Norwegian, Croatian, and Slovak languages.

Task	Run	Accuracy	F1 (micro)	F1 (weighted)
ADI	1	69.71	69.71	69.75
ADI	2	57.44	57.44	56.90
DSL	1	92.49	92.49	92.45
GDI	1	65.28	65.28	62.64

Table 5: Main results of language identification tasks on the test set as calculated by the organizers.

4 Results

4.1 Language identification

In the language identification subtasks, our best performing models were SVM models with the parameters listed in Table 4. We have participated in the shared task using only these models. For the ADI task, we submitted two runs, the first one using both the transcriptions and the i-vectors, and the second one using only the transcriptions. The scores of our systems in each task on the test set is presented in Table 5.

According to rankings based on absolute F1 scores, our results indicate that the systems are in mid-range in all tasks. More precisely, we get 4th, 3th, 6th, positions in DSL, ADI, and GDI tasks, respectively. However, for the DSL task, the difference from the best score is rather small. Our accuracy scores are behind the top scores in each task by 0.25 %, 6.57 % and 2.78 % for DSL, ADI, and GDI respectively. We also present the confusion matrices for each task. For the DSL task, as shown in Table 6, almost all confusions occur within the groups. Within the groups, there seems to be a slight tendency for the members of the group with shorter documents on average to be confused more. Looking at inter-language group confusions on the development set more closely reveals that all such confusions are difficult to classify correctly without further context. Table 9 lists a few of the documents that were assigned a label from another language group by the classifier. The confused documents mainly consist of named entities, addresses, numbers or other symbols.

The confusion tables for ADI and GDI tasks are presented in Table 7 and Table 8 respectively. Since these represent a single group of varieties, the confusions are common in both tables. We do not observe any clear patterns in the mistakes made by the classifier in ADI task. Similarly, the confusion matrix of the GDI task does not indicate very clear patterns, except the Lucerne variety seems to be very difficult to identify for our system. The documents from the Lucerne area are more often recognized as from Basel or Zurich than Lucerne itself.

In our last year's participation, we only used character n-grams as features. Intuitively, the character n-grams are useful since they can capture parts of the morphology of languages. This helps generalizing over suffixes or prefixes that were possibly not observed in the training data. Larger character n-grams also include words, and also fragments from word sequences. However, very large character n-grams do not provide much help since they suffer from data sparsity. In our experiments, we often found improvements in language discrimination up to 7-grams. This may not be able to capture most variety-specific word bigrams or trigrams. As a result, we expect word n-grams to be also useful, despite the fact the information from (large) character n-grams and word n-grams will overlap considerably. To investigate the relative merits of combining character and word ngrams, we present the best average accuracies scores obtained with 10-fold cross validation experiments on the DSL training and development set combination in Table 10. Increasing the maximum length of the character n-grams helps in for all cases up to character n-gram length of 7. Increasing maximum word n-grams length also has a positive effect in all cases, although, the effect diminishes after bigrams.

As in the previous year, the accuracy of the neural network model was close to the SVM model, but despite additional efforts of tuning, the neural models did not perform better than the SVM model in any of the tasks. We performed a random search involving the type of feature extractors for characters and words, the length of embeddings for characters and words, the width of the convolutional filter (in case one of the feature extractors were convolutional networks), length of the embedding representations (number of convolutions, or length of RNN representations), and the amount of dropout used in various parts of the network.

In the case of the DSL development set, the best accuracy score obtained by the neural network was 90.72 as opposed 92.58 from our best performing SVM model in the same setting. In general, the performance of the model was relatively stable across 200 different random configurations of hyperparameters listed above, all lying within the range 0.88–0.91. Convolutional networks performed well over characters, but they yielded bad scores over the words, likely due to large number of filters over words that would be needed in the multilingual corpus processing. Recurrent neural network flavors (GRUs and LSTMs) were among the better options for obtaining better document representations from the word embeddings. However, simple averaging of the embedding vectors performed similarly. On character features, recurrent networks were impractical in our computing environment due to longer input sequence (512 characters).

4.2 Cross-lingual parsing

We used UDpipe (Straka et al., 2016) to train our parsers on the translated treebanks. We report both the Labeled Attachment Scores (LAS) and the Unlabeled Attachment Scores (UAS) in Table 11. In the case of Norwegian, we trained our system on both individual and combined treebanks from Swedish and Danish. In the case of Norwegian, we obtained the best results (9 points more than the baseline) when we trained the dependency parser on Norwegian treebank which is translated from Swedish. We obtained slightly better results than the baseline in the case of Croatian. In the case of Slovak, we obtained an improvement of 10 points over the baseline. In all the cases, our results are behind the other two participants by a margin of 5 points in Croatian and Norwegian; and, 14 points in the case of Slovak.

5 Discussion and conclusions

In this paper we described our systems participating in the VarDial 2017 shared tasks. We participated in all the four tasks offered during this shared task campaign. Although our main focus has been language identification tasks, we have also participated in the cross-lingual parsing shared task with a simple approach, and reported results in this paper.

Our participation in the language discrimination tasks, namely *Discriminating between similar languages* (DSL), *Arabic dialect identification* (ADI), and *German dialect identification* (GDI), is similar to to our previous year's participation (Çöltekin and Rama, 2016). We experimented with both SVMs and (deep) neural network models. Similar to our last year's experience, SVMs performed better than neural networks. This is inline with

	hr	bs	sr	es-ar	es-es	es-pe	fa-af	fa-ir	fr-ca	fr-fr	id	my	pt-br	pt-pt
hr	873	112	13	0	0	0	0	0	0	1	0	0	1	0
bs	112	783	103	0	1	0	0	0	0	1	0	0	0	0
sr	8	64	927	0	0	0	0	0	0	1	0	0	0	0
es-ar	0	0	0	836	62	93	0	0	0	3	0	0	4	2
es-es	0	0	0	72	879	45	0	0	0	0	0	0	2	2
es-pe	0	0	0	18	28	953	0	0	0	1	0	0	0	0
fa-af	0	0	0	0	0	0	969	31	0	0	0	0	0	0
fa-ir	0	0	0	0	0	0	31	968	0	0	0	0	1	0
fr-ca	0	0	0	0	0	0	0	0	951	49	0	0	0	0
fr-fr	0	0	0	0	0	0	0	0	61	939	0	0	0	0
id	0	0	0	0	2	0	0	0	0	1	983	14	0	0
my	0	0	0	0	0	0	0	0	0	2	10	88	0	0
pt-br	0	0	0	0	0	0	0	0	0	2	0	0	950	48
pt-pt	0	0	0	0	1	0	0	0	0	1	0	0	49	949

Table 6: Confusion matrix for the DSL task.

	egy	glf	lav	msa	nor
egy	210	18	37	17	20
glf	15	165	45	13	12
lav	36	40	218	17	23
msa	10	16	10	212	14
nor	36	22	36	15	235

Table 7: Confusion matrix for the ADI task.

	be	bs	lu	zh
be	634	57	24	191
bs	69	679	41	150
lu	181	263	244	228
zh	21	27	11	818

Table 8: Confusion matrix for the GDI task.

gold std.	predicted	text
hr	fr-FR	`2. 27/4 vrt 118 27/2 149,60`
fr-FR	hr	`Nadal (Esp) { Cilic (Cro): 6-2, 6-4, 6-3`
bs	fr-FR	`- 17.30 Galatasaray - Jadran (Split)`
pt-BR	fr-FR	`Shangri-La: 10 Avenue d'Iéna, 16ème arrondissement, Paris. Tel. (33 1) 5367-1998.`
id	pt-BR	`Kiper: Julio Cesar (Inter Milan), Victor (Gremio), Jefferson (Botafogo), Fabio (Cruzeiro)`

Table 9: Examples of inter-group confusions from the DSL task.

		Max word n-gram length			
		0	1	2	3
Max char n-gram length	0		90.47	91.48	91.53
	1	84.25	90.84	91.63	91.84
	2	84.25	91.68	92.07	92.22
	3	90.16	91.83	92.23	92.24
	4	91.69	92.12	92.38	92.40
	5	92.17	92.37	92.49	92.53
	6	92.34	92.48	92.55	92.55
	7	92.39	92.50	92.56	92.58
	8	92.37	92.48	92.52	92.54

Table 10: Best accuracy scores obtained on the DSL data by combinations of character and word n-grams of varying sizes.

target (source)	Baseline		Translation	
	LAS	UAS	LAS	UAS
no (sv)	56.63	66.24	65.62	74.61
no (da)	54.91	64.53	58.55	67.48
no (sv+da)	59.95	69.02	64.91	73.50
hr (sl)	53.35	63.94	55.20	66.75
sk (cz)	53.72	65.70	64.05	73.16

Table 11: Labeled (LAS) and unlabeled (UAS) attachment scores obtained by the translation model in comparison to the baseline provided by the organizers.

the results of VarDial 2016 shared task, where linear models (Jauhiainen et al., 2016; Zirikly et al., 2016; Goutte and Léger, 2016; Herman et al., 2016; Cianflone and Kosseim, 2016; Barbaresi, 2016; Adouane et al., 2016; McNamee, 2016; Nisioi et al., 2016; Gamallo et al., 2016; Malmasi and Zampieri, 2016; Ionescu and Popescu, 2016; Eldesouki et al., 2016, for example), performed better than the neural network models (Bjerva, 2016; Belinkov and Glass, 2016). Our current experiments also follow the same trend. As in the last year, our SVM models performed better than neural network models, and our main results only include scores obtained by SVM classifiers.

Unlike last year, where we only used character n-grams, this year we used a combination of character and word n-grams as features, and tuned the maximum number of n-grams included for each task. We obtained scores competitive with the scores of the other participating teams. In general, all scores are slightly higher for the DSL task compared to the last year. Besides the results on the shared task, we presented some results from the additional experiments that we performed in Section 4. The combination of character and word n-grams seem to have made a small but consistent difference in the experiments performed on the development data.

For the cross-lingual parsing task, we followed a simple method by automatically translating the source treebank and training an off-the-shelf parser on the translated treebank. We did not perform any further adaptation or pre-trained word representations which may have been helpful in this task. Although we obtained results that are consistently better than the baseline, our results have been substantially lower than the scores of the other two participating systems.

Acknowledgments

The authors thank the reviewers for the comments which helped improve the paper. The second author is supported by the ERC Advanced Grant 324246 EVOLAEMP, which is gratefully acknowledged.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Wafia Adouane, Nasredine Semmar, and Richard Johansson. 2016. ASIREM Participation at the Discriminating Similar Languages Shared Task 2016. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 163–169, Osaka, Japan.

Adrien Barbaresi. 2016. 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.

Yonatan Belinkov and James Glass. 2016. A Character-level Convolutional Neural Network for Distinguishing Similar Languages and Dialects. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 145–152, Osaka, Japan.

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.

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.

François Chollet. 2015. Keras. `https://github.com/fchollet/keras`.

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.

Mohamed Eldesouki, Fahim Dalvi, Hassan Sajjad, and Kareem Darwish. 2016. QCRI @ DSL 2016: Spoken Arabic Dialect Identification Using Textual Features. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 221–226, Osaka, Japan.

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.

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.

Cyril Goutte and Serge Léger. 2016. Advances in Ngram-based Discrimination of Similar Languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 178–184, Osaka, Japan.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating Similar Languages: Evaluations and Explorations.

Ondřej Herman, Vít Suchomel, Vít Baisa, and Pavel Rychlý. 2016. DSL Shared Task 2016: Perfect Is The Enemy of Good Language Discrimination Through Expectation–Maximization and Chunk-based Language Model. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 114–118, Osaka, Japan.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. 11(3):311–325.

Radu Tudor Ionescu and Marius Popescu. 2016. UnibucKernel: An Approach for Arabic Dialect Identification Based on Multiple String Kernels. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 135–144, Osaka, Japan.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2016. HeLI, a Word-Based Backoff Method for Language Identification. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 153–162, Osaka, Japan.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall, second edition.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Shervin Malmasi and Marcos Zampieri. 2016. Arabic Dialect Identification in Speech Transcripts. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 106–113, Osaka, Japan.

Shervin Malmasi, Eshrag Refaee, and Mark Dras. 2015. Arabic Dialect Identification using a Parallel Multidialectal Corpus. In *Proceedings of the 14th Conference of the Pacific Association for Computational Linguistics (PACLING 2015)*, pages 209–217, Bali, Indonesia, May.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72. Association for Computational Linguistics, July.

Paul McNamee. 2016. Language and Dialect Discrimination Using Compression-Inspired Language Models. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 195–203, Osaka, Japan.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 629–637. Association for Computational Linguistics, July.

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.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 23–28.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146, October.

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.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Ud-pipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Jörg Tiedemann and Nikola Ljubešić, Ljubešić. 2012. Efficient discrimination between closely related languages. In *Proceedings of COLING 2012*, pages 2619–2634.

Jörg Tiedemann, Željko Agić, and Joakim Nivre. 2014a. Treebank translation for cross-lingual parser induction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 130–140. Association for Computational Linguistics, June.

Jörg Tiedemann, Željko Agić, and Joakim Nivre. 2014b. Treebank translation for cross-lingual parser induction. In *Eighteenth Conference on Computational Natural Language Learning (CoNLL 2014)*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Jörg Tiedemann. 2014. Rediscovering annotation projection for cross-lingual parser induction. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1854–1864. Dublin City University and Association for Computational Linguistics, August.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pages 1–8. Association for Computational Linguistics.

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 Ljubešic, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the dsl shared task 2015. In *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)*, Valencia, Spain.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42.

Ayah Zirikly, Bart Desmet, and Mona Diab. 2016. The GW/LT3 VarDial 2016 Shared Task System for Dialects and Similar Languages Detection. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 33–41, Osaka, Japan.