Natural Language Engineering (2019), **25**, pp. 607–632
doi:10.1017/S1351324919000299

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Combining *n*-grams and deep convolutional features for language variety classification

Matej Martinc[1]* and Senja Pollak[1,2]

[1]Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia and
[2]Usher Institute of Population Health Sciences and Informatics, Edinburgh Medical School, Usher Institute, University of Edinburgh, Edinburgh, UK
*Corresponding author. Email: matej.martinc@ijs.si

(Received 05 November 2018; revised 12 April 2019; accepted 12 May 2019; first published online 18 July 2019)

Abstract

This paper presents a novel neural architecture capable of outperforming state-of-the-art systems on the task of language variety classification. The architecture is a hybrid that combines character-based convolutional neural network (CNN) features with weighted bag-of-*n*-grams (BON) features and is therefore capable of leveraging both character-level and document/corpus-level information. We tested the system on the Discriminating between Similar Languages (DSL) language variety benchmark data set from the VarDial 2017 DSL shared task, which contains data from six different language groups, as well as on two smaller data sets (the Arabic Dialect Identification (ADI) Corpus and the German Dialect Identification (GDI) Corpus, from the VarDial 2016 ADI and VarDial 2018 GDI shared tasks, respectively). We managed to outperform the winning system in the DSL shared task by a margin of about 0.4 percentage points and the winning system in the ADI shared task by a margin of about 0.2 percentage points in terms of weighted F1 score without conducting any language group-specific parameter tweaking. An ablation study suggests that weighted BON features contribute more to the overall performance of the system than the CNN-based features, which partially explains the uncompetitiveness of deep learning approaches in the past VarDial DSL shared tasks. Finally, we have implemented our system in a workflow, available in the ClowdFlows platform, in order to make it easily available also to the non-programming members of the research community.

Keywords: language variety; author profiling; text classification; convolutional neural network; bag-of-n-grams

1. Introduction

Author profiling (AP), which deals with learning about the demographics of a person based on the text she or he produced, is becoming a strong trend in the field of natural language processing (NLP). Tasks such as age, gender, and language variety prediction (automatic distinction between similar dialects or languages) are becoming increasingly popular, in part also because of the marketing potential of this research. Most AP research communities are centered around a series of scientific events and shared tasks on digital text forensics, the two most popular being the evaluation campaign VarDial (Varieties and Dialects)[a] (Zampieri *et al.* 2014), focused on tasks related to the study of linguistic variation, and an event called PAN (Uncovering Plagiarism, Authorship, and Social Software Misuse)[b], which first took place in 2011 and was followed by a series of shared tasks organized since 2013 (Rangel *et al.* 2013).

[a]http://corporavm.uni-koeln.de/vardial/sharedtask.html
[b]http://pan.webis.de/

Table 1. Winning systems for AP classification tasks in PAN AP and VarDial DSL shared tasks (language variety tasks in bold)

Year	VarDial (DSL – closed track)	PAN (AP)
2014	**SVM + BON** (Goutte, Léger, and Carpuat (2014))	LIBLINEAR[5] + BON (López-Monroy *et al.* 2014)
2015	**SVM + BON** (Malmasi and Dras 2015)	LIBLINEAR[5] + BON (Alvarez-Carmona *et al.* 2015)
2016	**SVM + BON** (Çöltekin and Rama 2016)	SVM + BON (Vollenbroek *et al.* 2016)
2017	**SVM + BON** (Bestgen 2017)	**SVM + BON** (Basile *et al.* 2017)

While deep learning approaches are gradually taking over different areas of NLP, the best approaches to AP still use more traditional classifiers and require extensive feature engineering (Rangel, Rosso, Potthast *et al.* 2017). This fact can be clearly seen if we look at the architectures used by the teams winning the AP shared tasks in recent years. Table 1 presents the winning approaches to the VarDial Discriminating between Similar Languages(DSL) shared tasks and PAN AP (gender, age, personality, and language variety prediction) tasks between 2014 and 2017[c]. In fact, six out of eight winning teams used one or an ensemble of Support Vector machine (SVM) classifiers and bag-of-*n*-grams (BON) features[d] for classification (two other winning teams used a LIBLINEAR classifier[e] and BON features), and when it comes to the task of DSL (all VarDial DSL tasks and PAN 2017 AP task), SVM classifiers with BON features have been used by all of the winning teams. The best ranking system that employed a deep learning architecture was developed by Miura *et al.* (2017) and ranked fourth in the PAN 2017 AP shared task.

The main contribution of this paper is to demonstrate that it is possible to build a neural architecture capable of achieving state-of-the-art results in the field of AP, and more specifically on the task of DSL. The proposed neural system is unique in a sense that it combines sophisticated feature engineering techniques used in traditional approaches to text classification with the newer neural automatic feature construction in order to achieve synergy between these two feature types. Experiments were conducted on eight distinct language varieties. First, we report results on the DSL Corpus Collection (DSLCC) v4.0 (Tan *et al.* 2014) used in VarDial 2017 (Zampieri *et al.* 2017), which was chosen because of its size (with 294,000 documents it is by far the largest corpus used in the presented shared tasks) and because it contains six different language groups, which also allows to explore the possibility of building a generic architecture that would discriminate well between languages in many different language groups without any language-group-specific parameter tweaking. Second, we report results on two much smaller corpora, the Arabic Dialect Identification Corpus (ADIC) used in a VarDial 2016 ADI shared task (Malmasi *et al.* 2016*b*) and the German Dialect Identification Corpus (GDIC) used in a VarDial 2018 GDI shared task (Zampieri *et al.* 2018) in order to determine how data set size and characteristics affect the competitiveness of the proposed system. Finally, we want to encourage the reproducibility of results and offer a larger research community (including linguists and social scientists) an easy out-of-the-box way of using our system. Therefore, we have not only published our code online (`http://source.ijs.si/mmartinc/NLE_2017`) but also implemented the architecture in the clowd-based visual programming system ClowdFlows (Kranjc, Podpečan, and Lavrač (2012)).

[c]VarDial evaluation campaign 2018 was not included because there was no DSL shared task. PAN 2018 gender classification task is not included because the gender classification task dealt with determining the gender of the author from both text and image data.

[d]The term BON features is used in a broader sense here, covering features such as bag-of-words, character, and word BON and bag-of-part-of-speech tags.

[e]It is unclear from the system description papers by López-Monroy *et al.* (2014) and Alvarez-Carmona *et al.* (2015) whether linear SVM or logistic regression classifier was used.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

The paper is structured as follows. Section 2 addresses the related work on text classification in the field of AP. Section 3 describes the architecture of the proposed neural classification system in detail, while in Section 4 we report on our experimental setup. Results of the experiments and an error analysis are presented in Section 5, followed by an ablation study in Section 6. Section 7 presents the implementation of our approach in the ClowdFlows platform and finally, the conclusions and directions for further work are presented in Section 8.

2. Related work

The most popular approach to language variety classification usually relies on BON features and SVM classifiers (see Table 1). Bestgen (2017), the winner of the VarDial 2017 DSL task, used an SVM classifier with character *n*-grams, capitalized word character *n*-grams, *n*-grams of part-of-speech (POS) tags, and global statistics (proportions of capitalized letters, punctuation marks, spaces, etc.) features. *N*-grams had sizes from one to seven and different feature configurations were used for different language groups. The novelty of this approach was the use of the BM25 weighting scheme (Robertson and Zaragoza 2009) instead of the traditional term frequency-inverse document frequency (TF-IDF). BM25 (also called Okapi BM25) is a version of TF-IDF with some modifications made to each of the two components (term frequency and inverse document frequency) that, most importantly, allow it to take into account the length of the document. The classical TF-IDF formula is

$$TF - IDF = tf * log(\frac{N}{df})$$

where *tf* is the number of terms in the document, *N* is the number of documents in the corpus, and *df* the number of documents that contain the term. On the other hand, the formula for BM25 is the following:

$$BM25 = \frac{tf}{tf + k_1 * (1 - b + b * \frac{dl}{dl - avg_{dl}})} * log(\frac{N - df + 0.5}{df + 0.5})$$

where k_1 is a free parameter for tuning the asymptotic maximum of the term frequency component of the equation, *dl* is a document length, avg_{dl} an average length of a document in the corpus, and *b* a free parameter for fine-tuning the document length normalization part of the equation. While Bestgen (2017) showed in his experiments that the choice of the weighting scheme does impact the performance of the classifier, the general employment of different weighting schemes by the best performing systems in past shared tasks (Zampieri *et al.* 2017) suggests that feature weighting in general is positively correlated with gains in classification performance.

A very similar SVM-based system but with simpler features (just word unigrams, bigrams and, character three- to five-grams) was used by the winners of the PAN 2017 competition Basile *et al.* (2017). The authors of the paper also discovered that adding more complex features into the model actually negatively affected its performance. An SVM ensemble with almost identical features (word unigrams and character one- to six-grams) was also used by the winners of the VarDial 2016 ADI task Malmasi *et al.* (2016*a*). An even more minimalistic SVM-based approach was proposed by the winners of the VarDial 2016 DSL competition (Çöltekin and Rama 2016), who used only character three- to seven-grams as features. The authors also report on the failed attempt to build two neural networks capable of beating the results achieved by the SVM, first one being the FastText model proposed by Joulin *et al.* (2016) and the second one a hierarchical model based on character and word embeddings. Another attempt of tackling the task with a neural approach was reported by Criscuolo and Aluisio (2017). They ranked ninth with a hybrid configuration composed of a word-level multi-layer-perceptron model and a character-level Naive Bayes model. They also experimented with a word-level convolutional neural network (CNN), which performed slightly worse than their hybrid classifier.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

There have also been some quite successful attempts of tackling the language variety prediction with neural networks. Miura *et al.* (2017) ranked fourth in the PAN 2017 shared task by using a system consisting of a recurrent neural network layer, a CNN layer, and an attention mechanism. In a set of VarDial 2018 evaluation campaign tasks, Ali tackled the tasks of distinguishing between four different Swiss German dialects (Ali 2018*a*), five Arabic dialects (Ali 2018*b*), and five closely related languages from the Indo-Aryan language family (Ali 2018*c*), ranking second in the first and second task and fourth in the third task, respectively. The system is based on character-level CNNs and recurrent networks. The one-hot encoded input sequence of characters enters the network through the recurrent GRU layer used as an embedding layer. Next is the convolutional layer with different filter sizes, ranging from two to seven, which is followed by a batch normalization, max-pooling, dropout, and finally a softmax layer used for calculating the probability distribution over the labels.

While neural networks were not a frequent choice in VarDial DSL 2017 (Zampieri *et al.* 2017), in the VarDial DSL 2016 shared task (Malmasi *et al.* 2016*b*) three teams used some form of CNN. Belinkov and Glass (2016) used a character-level CNN and ranked sixth out of seven teams, achieving more than six percentage points lower accuracy than the winning system. A somewhat more sophisticated system was employed by Bjerva (2016), who combined CNN with recurrent units, developing a so-called residual network that takes as input sentences represented at a byte level. He ranked fifth in the competition. A third team called *Uppsala* used a word-level CNN but did not submit a report about their approach.

Two rear occasions when an SVM-based system did not win in a language variety classification shared task occurred at VarDial 2018 GDI and Indo-Aryan Language Identification (ILI) tasks, where Jauhiainen *et al.* beat the nearest competition by a large margin of four percentage points (Jauhiainen, Jauhiainen, and Lindén (2018a)) and more than five percentage points (Jauhiainen, Jauhiainen, and Lindén (2018b)), respectively. Their Helsinki language identification (HeLI) method with adaptive language modeling was in both cases calculated on character four-grams. The HeLI system was, however, outperformed by a margin of almost five percentage points at the VarDial 2018 Discriminating between Dutch and Flemish in Subtitles task by an SVM-based system proposed by Çöltekin, Rama, and Blaschke (2018).

3. System architecture

Research presented in Section 2 indicates that using character-level CNNs might be the most promising neural approach to the task of DSL. CNNs are able to identify important parts of a text sequence by employing a max-over-time pooling operation (Collobert *et al.* 2011), which keeps only the character sequences with the highest predictive power in the text. These sequences of predefined lengths resemble character *n*-grams, which were used in nearly every winning approach in the past shared task, but the CNN approach also has the advantage over the traditional BON approaches that it preserves the order in which these text areas with high predictive power appear in the text.

On the other hand, its main disadvantage could be the lack of an effective weighting scheme that would be capable of determining how specific these character sequences are for every input document. The data are fed into a neural classifier in small batches; therefore, it is impossible for it to obtain a somewhat global view on the data and its structure, which is encoded in the more traditional TF-IDF (or BM25) weighted input matrix. Another intuition that might explain the usefulness of weighting schemes for the specific task of language variety classification is related to named entities, for which it was shown in the past shared tasks that they in many cases reflect the origin of the text (Zampieri *et al.* 2015). The hypothesis is that these entities are quite rare and somewhat document specific and are therefore given large weights by different weighting schemes, encouraging the classifier to pay attention to them. The importance of choosing an effective weighting scheme on the task of DSL is also emphasized in the research by Bestgen

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

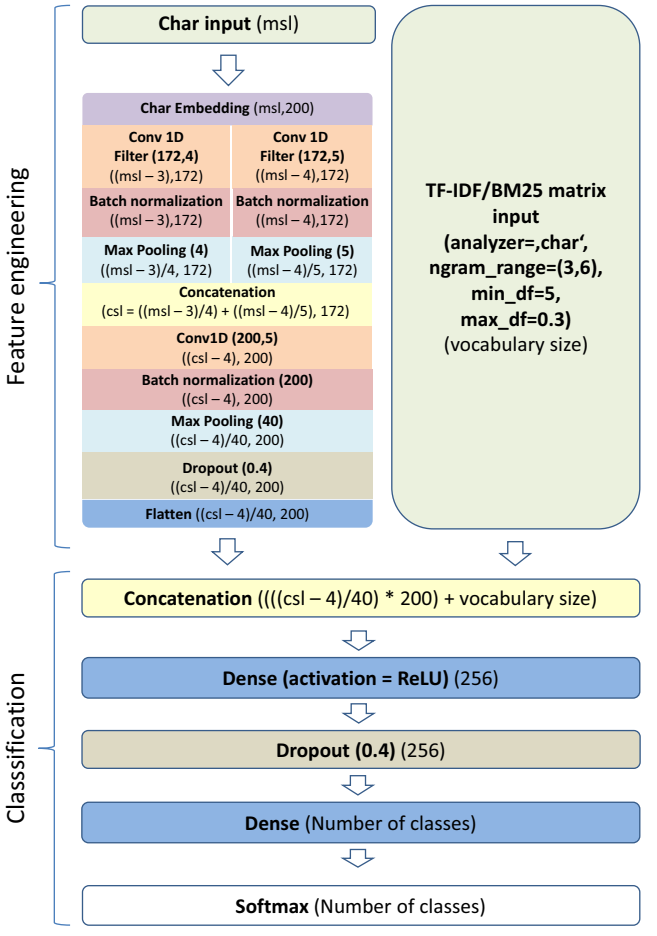

Figure 1. System architecture: layer names and input parameters are written in bold, layer output sizes are written in normal text, *msl* stands for maximum sequence length, and *csl* stands for concatenated sequence length.

(2017), the winner of the VarDial 2017 DSL task, who managed to gain some performance boost by replacing the TF-IDF weighting scheme with BM25.

Our architecture (visualized in Figure 1) builds on these findings from the literature and is in its essence an effective hybrid between a traditional feature engineering approach, which relies on different kinds of BON features, and a newer neural feature engineering approach to text classification. This combination of two distinct text classification architectures is capable of leveraging character-level and more global document/corpus-level information and achieving synergy between these two data flows. The main idea is to improve on standard CNN approaches by adding an additional input to the network that would overcome the lack of an effective weighting scheme. Therefore, the text is fed to the network in the form of two distinct inputs (as presented in Figure 1):

- *Char input*: Every document is converted into a numeric character sequence (every character is represented by a distinct integer) of length corresponding to the number of characters in the longest document in the train set (zero value padding is added after the document character sequence and truncating is also performed at the end of the sequence if the document in the validation or test set is too long).
- *TF-IDF/BM25 matrix*: We explore the effect of two distinct weighting schemes on the performance of the classifier; therefore, input data set is converted into a matrix of either TF-IDF

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

or BM25 weighted features with a *TfidfVectorizer* from ScikitLearn (Pedregosa *et al.* 2011) or our own implementation of the *BM25Vectorizer*. The matrix is calculated on character *n*-grams of sizes three, four, five, and six with a minimum document frequency of five and appearing in at most 30% of the documents in the train set. Sublinear term frequency scaling is applied in the term frequency calculation when *TfidfVectorizer* is used and for BM25 weighting parameters b and k_1 are set to 0.75 and 1.2, respectively, same as in Bestgen (2017).

The architecture for processing *Char input* is a relatively shallow character-level CNN with randomly initialized embeddings of size $msl \times 200$, where *msl* stands for *maximum sequence length*. Assuming that w is a convolutional filter, b is a bias, and f a nonlinear function (a rectified linear unit (*ReLU*) in our case), a distinct character *n*-gram feature c_i is produced for every possible window of h characters $x_{i:i+h-1}$ in the document according to the convolutional equation:

$$c_i = f(w \cdot x_{i:i+h-1} + b)$$

In the first step, we employ two parallel convolutional layers (one having a window of size four and the other of size five), each of them having 172 convolutional filters. These layers return two feature maps of size $(msl - ws + 1) \times 172$, where *ws* is the window size. Batch normalization and max-over-time pooling operations (Collobert *et al.* 2011) are applied on both feature maps in order to filter out features with low predictive power. These operations produce two matrices of size $(msl - ws + 1)/mws \times 172$, where sizes of max-pooling windows (*mws*) correspond to convolution window sizes. Output matrices are concatenated and the resulting matrix is fed into a second convolutional layer with 200 convolutional filters and window size five. Batch normalization and max-over-time pooling are applied again and after that we conduct a dropout operation on the output of the layer, in which 40% of input units are dropped in order to reduce overfitting. Finally, the resulting output is flattened (changed from a two-dimensional to a one-dimensional vector) and passed to a *Concatenation* layer, where it is concatenated with the input *TF-IDF/BM25 matrix*. The resulting concatenation is passed on to a fully connected layer (*Dense*) with a *ReLU* activation layer and dropout is conducted again, this time on the concatenated vectors. A final step is passing the resulting vectors to a dense layer with a *Softmax* activation, responsible for producing the final probability distribution over language variety classes.

4. Experimental setup

This section describes the data sets and the methodology used in our experiments.

4.1 Data

All experiments were conducted on three corpora described in Table 2:

- **DSLCC v4.0 (Tan *et al.* 2014)**[f]: the corpus used in the VarDial 2017 DSL shared task. The corpus contains 294,000 short excerpts of news texts divided into 6 distinct language groups (Slavic, Indonesian and Malay, Portuguese, Spanish, French, and Farsi) and covering 14 language varieties in total: Bosnian, Croatian and Serbian; Malay and Indonesian; Persian and Dari; Canadian and Hexagonal French; Brazilian and European Portuguese; Argentine, Peninsular, and Peruvian Spanish. Each language contains 20,000 documents for training (out of which 2000 are to be used as a validation set) and 1000 for testing.
- **ADIC (Ali *et al.* 2015)**[g]: the corpus used in the VarDial 2016 ADI shared task. It contains transcribed speech in Modern Standard Arabic, Egyptian, Gulf, Levantine, and North African

[f]The corpus is publicly available at http://ttg.uni-saarland.de/resources/DSLCC/

[g]The corpus is publicly available at http://alt.qcri.org/resources/ArabicDialectIDCorpus/varDial_DSL_shared_task_2016_subtask2/

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

Table 2. DSLCC v4.0, ADIC and GDIC corpora

DSLCC v4.0					
Language/Variety	Class	Train inst.	Train tokens	Test inst.	Test tokens
Bosnian	bs	20,000	716,537	1000	35,756
Croatian	hr	20,000	845,639	1000	42,774
Serbian	sr	20,000	777,363	1000	39,003
Indonesian	id	20,000	800,639	1000	39,954
Malay	my	20,000	591,246	1000	29,028
Brazilian Portuguese	pt-BR	20,000	907,657	1000	45,715
European Portuguese	pt-PT	20,000	832,664	1000	41,689
Argentine Spanish	es-AR	20,000	939,425	1000	42,392
Castilian Spanish	es-ES	20,000	1,000,235	1000	50,134
Peruvian Spanish	es-PE	20,000	569,587	1000	28,097
Canadian French	fr-CA	20,000	712,467	1000	36,121
Hexagonal French	fr-FR	20,000	871,026	1000	44,076
Persian	fa-IR	20,000	824,640	1000	41,900
Dari	fa-AF	20,000	601,025	1000	30,121
Total		280.000	8,639,459	14,000	546,790
ADIC					
Egyptian	EGY	1578	85,000	315	13,000
Gulf	GLF	1672	65,000	256	14,000
Levantine	LAV	1758	66,000	344	14,000
Modern Standard	MSA	999	49,000	274	14,000
North African	NOR	1612	52,000	351	12,000
Total		7619	317,000	1540	67,000
GDIC					
Bern	BE	4956	35,962	1191	12,013
Basel	BS	4921	36,965	1200	9802
Lucerne	LU	4593	38,328	1186	11,372
Zurich	ZH	4834	36,919	1175	9610
Total		19,304	148,174	4,752	42,797

dialects. Speech excerpts were taken from a multi-dialectical corpus containing broadcast, debate and discussion programs from Al Jazeera. Altogether 7619 documents were used for training (out of which 10% were used for validation) and 1540 documents for testing.

- **GDIC (Samardzic, Scherrer, and Glaser (2016))**: the corpus used in the VarDial 2018 GDI shared task. Texts were extracted from the ArchiMob corpus of Spoken Swiss German[h], which contains 34 oral interviews with people speaking Bern, Basel, Lucerne, and Zurich Swiss German dialects. A total of 19,304 documents were used for training (out of which 10% were used for validation) and 4752 for testing.

[h]The ArchiMob corpus is publicly available at https://www.spur.uzh.ch/en/departments/research/textgroup/ArchiMob.html

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

4.2 Methodology

For experiments in the DSLCC v4.0 we chose to use a two-step approach, as first proposed by Goutte, Léger, and Carpuat (2014):

(1) The general classifier is trained to identify the language group for every specific document. For this step, the input TF-IDF/BM25 matrix is calculated only on the word bound character *n*-grams[i] of sizes three, four, and five with a minimum document frequency of five and appearing in at most 30% of the documents in the train set. This configuration produces a TF-IDF/BM25 matrix of smaller size than if the configuration for the TF-IDF/BM25 matrix, described in Section 3, was used. This size reduction was chosen because distinguishing between different language groups is not a difficult problem, therefore, this parameter reduction does not influence performance but it reduces the execution time.
(2) We train six different classification models, one for each language group. After being classified as belonging to a specific language group by the general classifier in Step 1, the documents are assigned to the appropriate classifier for predicting the final language variety.

Since NLP tools and resources such as POS taggers, pretrained word embeddings, word dictionaries, and tokenizers might not exist for some underresourced languages, we also believe that an architecture which does not require language-specific resources and tools, apart from the training corpus, might be more useful and easier to use in real-life applications. For this reason, our system does not require any additional resources and the conducted preprocessing procedure is light[j].

We show (see Section 5) that the proposed architecture is generic enough to outperform the winning approach of VarDial 2017 on all of the language groups without any language-group-specific parameter or architecture tweaking. In contrast, most of the approaches of the VarDial 2017 DSL shared task resorted to language-group-specific optimization, as getting even the slightest possible performance boost by employing this tactic was important due to the competitive nature of shared tasks.

For the experiments on the smaller ADIC and GDIC data sets, we use the same hyperparameter configuration and TD-IDF/BM25 features as for the six classification models for specific language groups in the DSLCC v4.0 corpus because we want to explore the relation between model performance and data set size. The hypothesis is that the performance of traditional SVM approaches would be less affected by smaller data set size than neural approaches.

We conducted an extensive grid search on the DSLCC v4.0 in order to find the best hyperparameters for the model. All combinations of the following hyperparameter values were tested before choosing the best combination, which is written in bold in the list below and presented in Section 3:

- Learning rates: 0.001, **0.0008**, 0.0006, 0.0004, 0.0002
- Number of parallel convolutions with different filter sizes: [3] [4], [3,4], **[4,5]**, [5,6], [6,7], [3,4,5], [4,5,6], [5,6,7], [3,4,5,6], [4,5,6,7], [3,4,5,6,7]
- Character embedding sizes: 100, **200**, 400
- Dense layer sizes: 128, **256**, 512
- Dropout values: 0.2, 0.3, **0.4**, 0.5
- Number of convolutional filters in the first convolution step: 156, **172**, 200
- Number of convolutional filters in the second convolution step: 156, 172, **200**

[i]Word-bound character *n*-grams are made only from text inside word boundaries, for example, a sequence *this is great* would produce a word-bound character 4-gram sequence *this, is__, grea, reat*, in which _ stands for empty space character.

[j]We only replace all email addresses in the text with *EMAIL* tokens and all URLs with *HTTPURL* tokens by employing regular expressions. Even if this might not be relevant to all of the corpora, we keep the preprocessing unchanged for all the settings.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

- Size of a max-pooling window in the second convolution step: 10, 20, **40**, 60
- BON n sizes: [3] [4] [3,4], [4,5], [5,6], [6,7], [3,4,5], [4,5,6], [5,6,7], **[3,4,5,6]**, [4,5,6,7], [3,4,5,6,7]
- Minimum document frequency of an n-gram in the TF-IDF/BM25 matrix: [2], **[5]**, [10]
- BM25 b parameter: 0.5, **0.75**, 1.0
- BM25 k_1 parameter: 1.0, **1.2**, 1.4

The hyperparameters, which influenced the performance of the network the most, were the learning rate, CNN filter sizes, size of the max-pooling window, BON n size, and a minimum document frequency of n-grams. Too many parallel convolutions, small sizes of the max-pooling window, and low minimum document frequency of n-grams showed tendency toward overfitting, especially when used together in combination. In general, we noticed quite a strong tendency toward overfitting no matter the hyperparameter combination, which could be to some extent the consequence of feeding a high-dimensional TF-IDF/BM25 matrix to the network, which greatly increases the number of network parameters. We noticed that a combination of a relatively small learning rate and a large dropout worked best to counter this tendency.

Another thing we noticed is that using exactly the same configurations of convolutional filter sizes and n-gram sizes negatively affected the performance, which was slightly improved when the configurations did not completely overlap. The hypothesis is that synergy between two data flows is less effective if the information in these two data flows is too similar. The validation set results did however show that configurations containing 4- and 5-grams and filter sizes of 4 and 5 in general worked better than other configurations for DSLCC v4.0 classification; therefore, these configurations were used in both data flows despite the overlap.

We use the Python Keras library (Chollet 2015) for the implementation of the system. For optimization, we use an Adam optimizer (Kingma and Ba 2015) with a learning rate of 0.0008. For each language variety in the DSLCC v4.0, the model is trained on the train set for 20 epochs and tested on the validation set after every epoch. The models trained on the ADIC and GDIC data sets are trained for 80 epochs due to longer convergence time on less data. The model with the best performance on the validation set is chosen for the test set predictions.

5. Results

First we present results on the DSLCC v4.0, which is (as it is the largest and covers the largest number of language varieties) the main focus of this study, then we present results on ADIC and GDIC and finally, we present findings of the error analysis conducted on the misclassified Slavic documents of the DSLCC v4.0 corpus.

5.1 Results on the DSLCC v4.0

Table 3 presents the results achieved by our neural classifier in comparison to the winner of the VarDial 2017 DSL shared task (Bestgen 2017) in terms of weighted F1, micro F1, macro F1, and accuracy measures.

The first step of the two-step classification approach, distinguishing between different language groups (*All-language groups (TF-IDF)* and *All-language groups (BM25)* rows in Table 3), proved trivial for the system, which achieved almost perfect weighted F1 score and misclassified only 27 documents out of 14,000 in the test set when TF-IDF weighting scheme was used and 29 documents when BM25 weighting scheme was used. If we look at the confusion matrices for language group classification (Figures 2 and 3), both models had most difficulties distinguishing between Spanish and Portuguese language groups. Ten Spanish texts were misclassified as Portuguese but on the other hand, only one Portuguese document was misclassified as Spanish when TF-IDF weighting scheme was used. With BM25 weights, the classifier misclassified nine

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

Table 3. Results of the proposed language variety classifier on the DSLCC v4.0 for different language groups, as well as for the discrimination between language groups (All-language groups). Also the results for all language varieties (All-language varieties) are provided, for which a comparison with the official VarDial 2017 winners is made. Results for both weighting schemes, TF-IDF and BM25, are reported separately

Language group (weighting)	F1 (weighted)	F1 (micro)	F1 (macro)	Accuracy
All-language groups (TF-IDF)	**0.9981**	**0.9981**	**0.9980**	**0.9981**
All-language groups (BM25)	0.9979	0.9979	0.9980	0.9980
Spanish (TF-IDF)	**0.9136**	**0.9140**	**0.9136**	**0.9140**
Spanish (BM25)	0.9042	0.9047	0.9042	0.9047
Slavic (TF-IDF)	0.8645	0.8650	0.8645	0.8650
Slavic (BM25)	**0.8752**	**0.8753**	**0.8752**	**0.8753**
Farsi (TF-IDF)	0.9685	0.9685	0.9685	0.9685
Farsi (BM25)	**0.9690**	**0.9690**	**0.9690**	**0.9690**
French (TF-IDF)	**0.9570**	**0.9570**	**0.9570**	**0.9570**
French (BM25)	0.9545	0.9545	0.9545	0.9545
Malay and Indonesian (TF-IDF)	0.9855	0.9855	0.9855	0.9855
Malay and Indonesian (BM25)	**0.9860**	**0.9860**	**0.9860**	**0.9860**
Portuguese (TF-IDF)	**0.9480**	**0.9480**	**0.9480**	**0.9480**
Portuguese (BM25)	0.9460	0.9460	0.9460	0.9460
All-language varieties (TF-IDF)	**0.9310**	**0.9312**	**0.9310**	**0.9312**
All-language varieties (BM25)	0.9304	0.9305	0.9304	0.9305
VarDial 2017 winner Bestgen (2017)	0.9271	0.9274	0.9271	0.9274

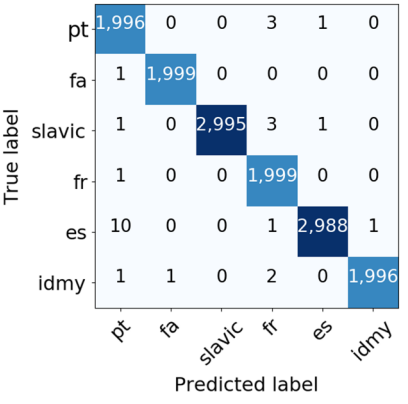

Figure 2. Confusion matrix for language group classification (TF-IDF weighting scheme).

Spanish documents as Portuguese and four Portuguese documents as Spanish. The analysis also reveals some surprising mistakes, such as that three Slavic documents and two documents from the Indonesian and Malay language group were misclassified as French with TF-IDF weighting and four documents from the Indonesian and Malay language group, three Spanish, and three French documents were classified as Slavic with BM25 weighting. A closer inspection of misclassified documents also reveals that these documents are in general much shorter (average word length is 9.74 and 10.17 when TF-IDF and BM25 are used respectively) than an average document in the Slavic sub-corpus (39.06 words long) and very likely contain some misleading named entities (e.g., a Slavic document, which was misclassifed as Spanish when TF-IDF weighting was used, contains the following text: *Caffe - Pizzeria ""BELLA DONNA"" u DOC-u*).

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

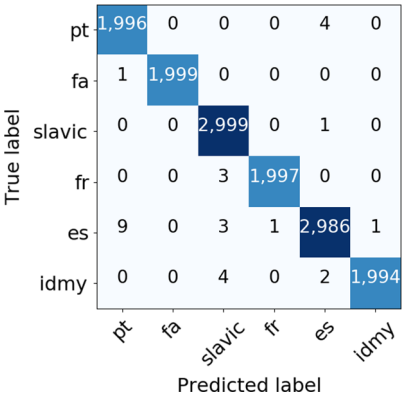

Figure 3. Confusion matrix for language group classification (BM25 weighting scheme).

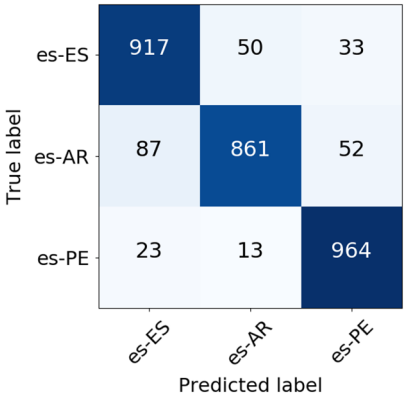

Figure 4. Confusion matrix for Spanish language varieties classification (TF-IDF weighting scheme).

Results for the second step of the two-step classification approach indicate that the difficulty of distinguishing language varieties within different language groups varies. The system had most difficulties with distinguishing between different Slavic languages, where it achieved by far the worst results with an weighted F1 of 0.8645 when TF-IDF weighting scheme was employed and about one percentage point better results when BM25 weighting was used. The second most difficult were Spanish varieties. We should point out that this comes as no surprise, since Slavic and Spanish languages groups were the only two groups that contained three varieties, while the other groups in DSLCC v4.0 contained two varieties. The system had least problems with distinguishing between Malay and Indonesian languages.

When it comes to comparing two weighting schemes, there is no clear overall winner. The biggest differences in performance are on Spanish varieties, where TF-IDF weighting outperforms BM25 by about one percentage point according to every measure, and on Slavic varieties, where BM25 weighting outperforms TF-IDF by a very similar margin. The differences on other varieties are smaller, ranging from 0.005 on Farsi and Malay and Indonesian varieties to 0.020 on Portuguese varieties.

Confusion matrices for specific language varieties enable a more thorough analysis of the results. For Spanish varieties (Figures 4 and 5), the system had most problems distinguishing between Argentine and Castilian Spanish. The second most common mistake no matter the weighting scheme was classifying Argentine Spanish as the Peruvian variety of Spanish. On the other hand, Peruvian Spanish was the easiest to classify by the system, with altogether only 36 (TF-IDF weighting) and 37 (BM25 weighting) misclassified instances.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

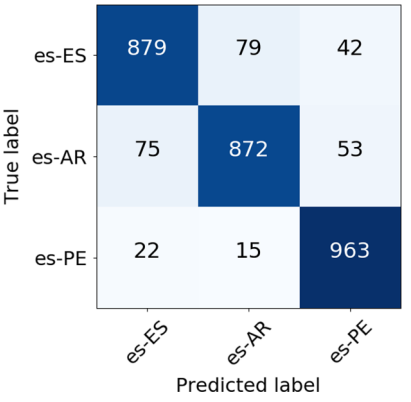

Figure 5. Confusion matrix for Spanish language varieties classification (BM25 weighting scheme).

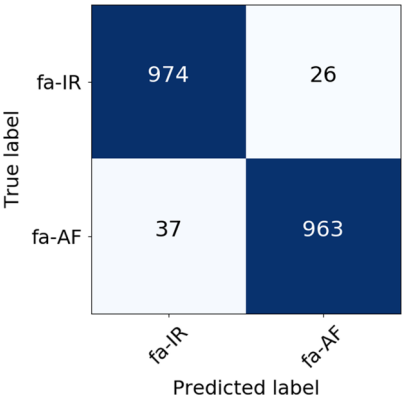

Figure 6. Confusion matrix for Farsi language varieties classification (TF-IDF weighting scheme).

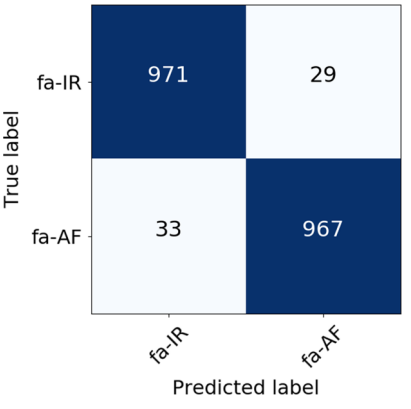

Figure 7. Confusion matrix for Farsi language varieties classification (BM25 weighting scheme).

The system performed well for all binary predictions (Figures 6 and 7, Figures 8 and 9, Figures 10 and 11, Figures 12 and 13) and the difference in performance between two weighting schemes are small. Out of these confusion matrices, the most unbalanced with regard to false predictions is the confusion matrix for Indonesian and Malay variety (Figure 10), where twice as many Indonesian documents were classified as Malay than the other way around when TF-IDF weighting was used. Although, as mentioned before, distinguishing between Indonesian and Malay was the least difficult task for the classifier and altogether only 29 and 28 instances were misclassified when TF-IDF and BM25 weighting were used, respectively.

For Slavic languages (Figures 14 and 15), the hardest problem for the system was distinguishing between Croatian and Bosnian, with 113 Bosnian documents being classified as Croatian and 112 Croatian documents being classified as Bosnian when TF-IDF weighting was used and with 113

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

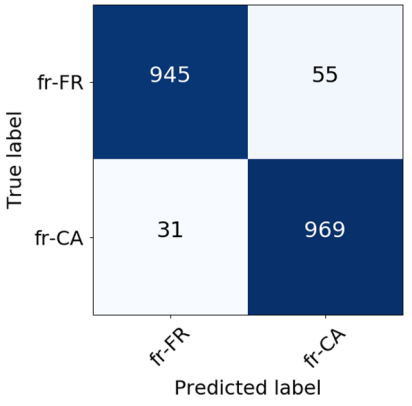

Figure 8. Confusion matrix for French language varieties classification (TF-IDF weighting scheme).

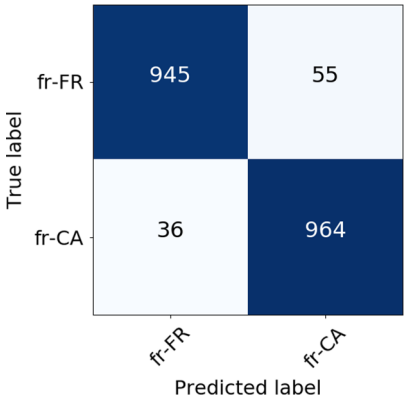

Figure 9. Confusion matrix for French language varieties classification (BM25 weighting scheme).

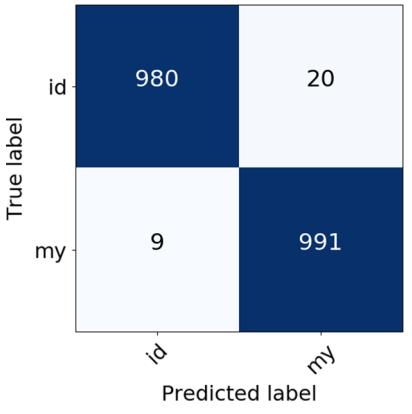

Figure 10. Confusion matrix for Indonesian and Malay variety classification (TF-IDF weighting scheme).

Bosnian documents being classified as Croatian and 99 Croatian documents being classified as Bosnian when BM25 weighting was employed. Distinguishing between Bosnian and Serbian was also not trivial for the classifier no matter the weighting scheme, with 94 Bosnian documents being misclassified as Serbian and 66 Serbian documents misclassified as Bosnian when TF-IDF weighting scheme was deployed and 73 Bosnian documents being misclassified as Serbian and vice versa when BM25 weighting was used. On the other hand, distinguishing between Serbian and Croatian is a much easier problem, with altogether only 20 (TF-IDF weighting) and 16 (BM25 weighting) documents being misclassified.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

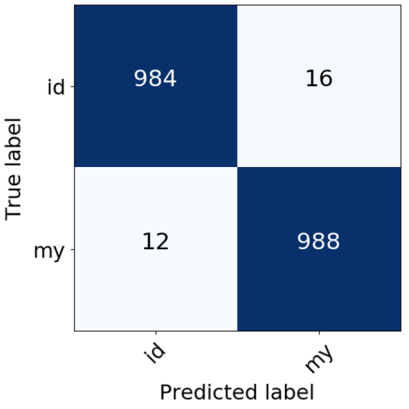

Figure 11. Confusion matrix for Indonesian and Malay variety classification (BM25 weighting scheme).

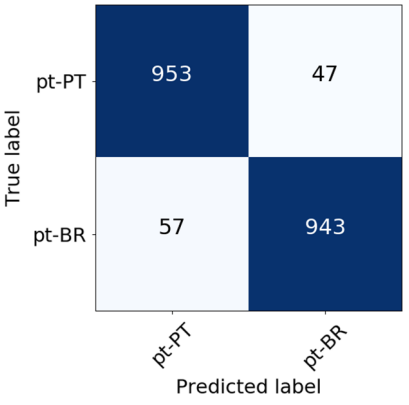

Figure 12. Confusion matrix for Portuguese language varieties classification (TF-IDF weighting scheme).

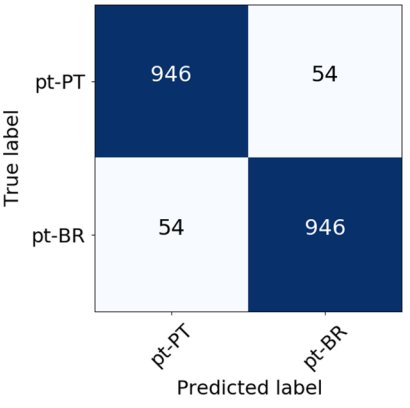

Figure 13. Confusion matrix for Portuguese language varieties classification (BM25 weighting scheme).

Overall (rows *All-language varieties (TF-IDF)* and *All-language varieties (BM25)* in Table 3), the neural network outperforms the SVM-based approach used by the winners of the shared task by about 0.4 percentage points according to all measures when TF-IDF weighting scheme is used. BM25 weighting performs slightly worse but still outperforms state of the art by about 0.35 percentage points margin. Our results therefore differ from the study conducted by Bestgen (2017), the winner of the shared task, where he reported improvement in performance for all but one language group when TF-IDF weighting is replaced by BM25. It should, however, be noted that these improvements were only reported on the validation set and no comparison between weighting schemes was done on the official test set.

There were no available reported results for individual language groups on the official test set, therefore we provide a comparison with the VarDial 2017 DSL winning team on the validation set,

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

Table 4. Accuracy comparison of our system to the VarDial 2017 DSL winners on validation sets

Language group (weighting)	Our system	VarDial 2017 winner	Improvement (%)
Spanish (TF-IDF)	**0.9180**	0.8970	2.10
Spanish (BM25)	**0.9202**	0.9030	1.72
Slavic (TF-IDF)	**0.8663**	0.8445	2.18
Slavic (BM25)	**0.8670**	0.8506	1.64
Farsi (TF-IDF)	**0.9685**	0.9598	0.87
Farsi (BM25)	**0.9720**	0.9632	0.88
French (TF-IDF)	**0.9588**	0.9396	1.92
French (BM25)	**0.9590**	0.9472	1.18
Malay and Indonesian (TF-IDF)	**0.9863**	0.9835	0.28
Malay and Indonesian (BM25)	**0.9875**	0.9827	0.48
Portuguese (TF-IDF)	**0.9440**	0.9299	1.41
Portuguese (BM25)	**0.9428**	0.9355	0.73

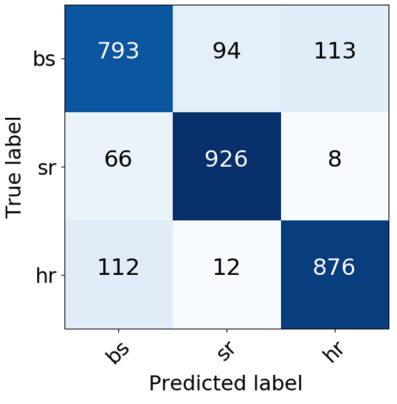

Figure 14. Confusion matrix for Slavic language varieties classification (TF-IDF weighting scheme).

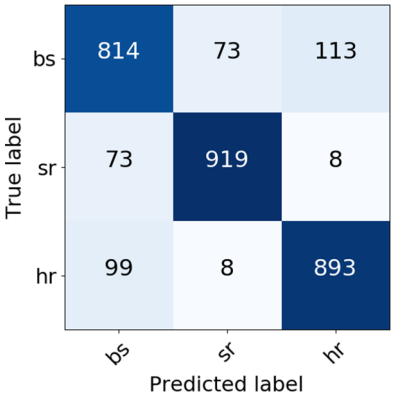

Figure 15. Confusion matrix for Slavic language varieties classification (BM25 weighting scheme).

as the author (Bestgen 2017) reports them when presenting the benefits of the weighting scheme BM25 (in their Table 3 on p. 119). Note, however, that the results report on a slightly simplified system, as for the weighting scheme comparison, the author used only character *n*-grams features. Comparison results are presented in Table 4.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

Table 5. Results of the proposed language variety classifier on the ADIC and GDIC. Results for both weighting schemes, TF-IDF and BM25, are reported separately

Language group (weighting)	F1 (weighted)	F1 (micro)	F1 (macro)	Accuracy
ADIC (TF-IDF)	**0.5152**	0.5123	0.5147	**0.5123**
ADIC (BM25)	0.5090	0.5097	0.5067	0.5097
VarDial ADI 2016 winner Malmasi *et al.* (2016*a*)	0.5132	/	/	0.5117
GDIC (TF-IDF)	0.6281	0.6294	0.6280	0.6294
GDIC (BM25)	0.6289	0.6311	0.6289	0.6311
VarDial GDI 2018 winner Jauhiainen *et al.* (2018*a*)	/	/	**0.6860**	/

Our system performs better than the simplified version of the VarDial 2017 DSL shared task winning system on all language groups. When TF-ID weighting is used by both systems, the differences vary from around two percentage points on Spanish, Slavic, and French language groups, to about 1.5 percentage point difference on the Portuguese language group, and finally, to only 0.28 percentage point difference on Malay and Indonesian, which are the easiest languages to distinguish for both of the classifiers. When BM25 weighting scheme is used, the differences are smaller, ranging from about 1.5 percentage point on Spanish and Slavic to about 0.5 percentage point on Malay and Indonesian.

Interestingly, when it comes to comparing both weighting schemes only on validation sets, the influence on the performance of our system when BM25 weighting is used is quite consistent with the influence reported by Bestgen (2017). By using BM25 weighting, the performance is improved on five out of six language groups, same as in Bestgen (2017), although the language groups are not the same: in Bestgen (2017) performance is not improved on the Malay and Indonesian language group while we report no improvement on Portuguese. However, these improvements at least in our case do not translate well to performance improvements on the official test set.

5.2 Results on ADIC and GDIC

Table 5 presents the results achieved by our neural classifier on the ADIC and GDIC corpora in comparison to the winners of the VarDial ADI 2016 and VarDial GDI 2018 shared tasks. The system manages to improve on the state of the art on the ADIC by a small margin of about 0.2 percentage point according to the weighted F1 score when TF-IDF weighting is used, even though the ADIC contains more than 10 times less documents per class than the language varieties in the DSLCC v4.0. By using BM25 weighting, the performance of the classifier is about 0.6 and 0.2 percentage points worse in terms of accuracy and weighted F1 score. On the other hand, the results on the GDIC are almost six percentage points lower than the current state-of-the-art HeLI method (Jauhiainen *et al.* 2018*a*) in terms of macro F1 score. Our system also performed worse than the SVM-based system proposed by Çöltekin *et al.* (2018) and a recurrent neural network proposed by Ali (2018a), which achieved macro F1 scores of 0.646 and 0.645, respectively. We can also observe that BM25 weighting slightly improves the performance according to all the criteria. Results on ADIC and GDIC corpora are somewhat in line with the initial hypothesis that neural approaches are more affected by a small data set size than more traditional SVM approaches. Previous SVM-based state of the art on the ADIC corpora is outperformed by a smaller margin than the DSLCC v4.0 state of the art and the proposed system performs worse than the second ranked SVM system (2018) on the GDIC corpus.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

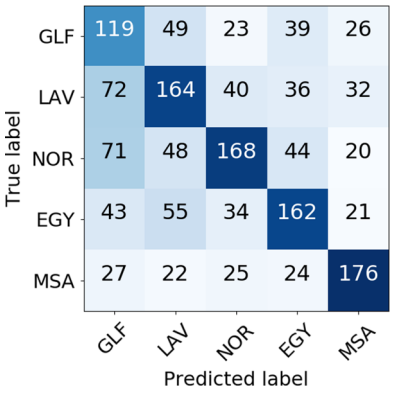

Figure 16. Confusion matrix for Arabic language varieties classification (TF-IDF weighting scheme).

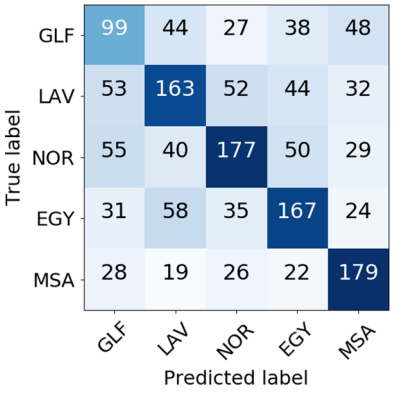

Figure 17. Confusion matrix for Arabic language varieties classification (BM25 weighting scheme).

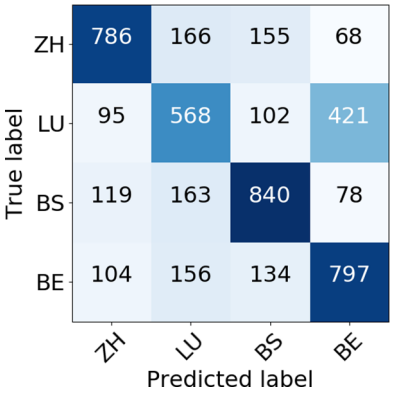

Figure 18. Confusion matrix for German language varieties classification (TF-IDF weighting scheme).

Confusion matrices for the ADIC (Figures 16 and 17) show that the Modern Standard Arabic is the easiest to classify no matter the weighting scheme. We can also see that if BM25 weighting is used, the classifier struggles much more with the Gulf dialect, correctly classifying only 99 out of 256 instances, than if TF-IDF weighting is used, in which case it correctly classifies 119 instances.

Confusion matrices for the GDIC (Figures 18 and 19) show that the choice of the weighting scheme does not have as big of an influence on the performance of the classifier as in the case of ADIC. No matter the weighting scheme, by far the most common mistake was misclassifying the Lucerne dialect as a Bern dialect. Interestingly, the opposite mistake of misclassifying Bern dialect

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

Table 6. Results of the error analysis on 405 misclassified Slavic documents

Group	Num. doc.	Prop. of doc.	Avg. doc. length
No named entities	144	0.36	26.94
Misleading named entities	70	0.17	40.96
Clarifying named entities	41	0.10	34.96
Unrelated named entities	150	0.37	33.17
All misclassified	405	1.00	32.48

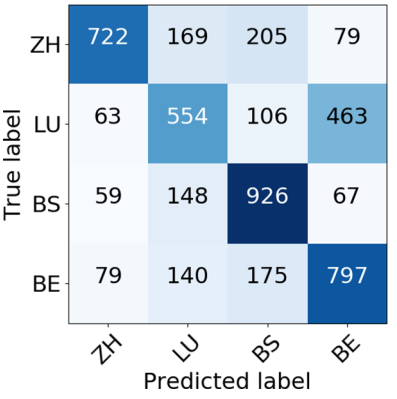

Figure 19. Confusion matrix for German language varieties classification (BM25 weighting scheme).

as Lucerne dialect is much rarer, which might be connected to some extent to the fact that the train set contains 328 more documents for the Bern dialect than for the Lucerne dialect.

5.3 Error analysis

We conducted a manual error analysis on the misclassified Slavic documents[k] in order to get a clearer picture about what kind of documents are the hardest to classify. Misclassified documents were manually grouped into four classes according to the number and type of named entities found in the document:

- **No named entities**: Documents without any named entities.
- **Misleading named entities**: Documents containing any named entities (e.g., names of regions, cities, public figures) originating from a country with the official language variety corresponding to one of the two possible incorrect language varieties (e.g., a document labeled as Serbian containing the word *Zagreb*, which is the capital of Croatia, would be put into this class).
- **Clarifying named entities**: Documents containing named entities originating from a country with the official language variety being the correct language variety and containing no misleading entities.
- **Unrelated named entities**: Documents containing only named entities that are not originating from any of the countries speaking target language varieties (e.g., a document containing only the named entity *Budapest* would be classified into this category).

Results of the analysis are presented in Table 6. Results show that a large portion of misclassified documents (73%) either contain no named entities (36%) or contain only unrelated named entities (37%), which might make them harder to classify, although we cannot claim that for sure, since we

[k]Error analysis was conducted on documents misclassified by the system that employed TF-IDF weighting scheme.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

Table 7. Results of the ablation study. Column *CNN F1 (weighted)* presents performance of the system in terms of weighted F1 if only CNN-based features are used, column *BON F1 (weighted)* presents performance of the system if only TF-IDF-weighted BON features are used and column *All F1 (weighted)* presents the performance when these two types of features are combined

Language group	All F1 (weighted)	CNN F1 (weighted)	BON F1 (weighted)
DSLCC v4.0			
All-language groups	**0.9981**	0.9971	0.9976
Spanish	**0.9136**	0.8599	0.8863
Slavic	**0.8645**	0.8300	0.8594
Farsi	**0.9685**	0.9465	0.9610
French	**0.9570**	0.9325	0.9420
Malay and Indonesian	0.9855	0.9560	**0.9875**
Portuguese	**0.9480**	0.8994	0.9434
All-language varieties	**0.9310**	0.8935	0.9199
ADIC	0.5152	0.3971	**0.5177**
GDIC	**0.6281**	0.6059	0.6190

do not know the distribution of these classes across the entire test set. About 17% of the documents on the other hand contain misleading named entities that could influence the classifier prediction. There are also 41 documents (10%) containing only clarifying named entities that would be easily classified correctly by any human annotator with some basic background knowledge about Serbia, Bosnia, and Croatia. This suggests that there is still some room for improvement for the developed classifier.

Another finding is that misclassified documents are in average shorter (32.48 words long) than an average document from a Slavic language group (39.18 words long), suggesting that shorter documents are harder to classify by the classifier due to less available information. We can also see that the only group containing documents with similar length as the whole test set are documents containing misleading named entities (40.96 words long), which suggests that the classifier does somewhat rely on named entities during the prediction process.

6. Ablation study

The main novelty of our approach is the combination of weighted BON features with CNN-generated character features in the neural architecture. We carried out an ablation study in order to determine the contribution of these two types of features in the overall performance. To measure the contribution of weighted BON features, we removed the part of the system that deals with the convolutional processing of the character sequence input (the left side of the feature engineering part sketched in Figure 1). On the other hand, we removed the TF-IDF/BM25 matrix input in order to determine the contribution of the CNN-generated character features. Only TF-IDF weighting was used in the ablation study. The results of the study are presented in Table 7.

In all cases, classifier with only TF-IDF-weighted BON features (BON classifier) performs better than the classifier with only CNN-based features (CNN classifier), which also raises questions about the established deep learning paradigm that in a large majority of cases relies only on the automatically generated neural features. In DSLCC v4.0, the difference in performance is the largest in the case of Portuguese language variety classification, measuring more than four percentage points. If we ignore the language group classification, which is apparently trivial for all

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

Table 8. Results of the error analysis on Slavic documents misclassified by the BON classifier and correctly classified by the CNN classifier and on Slavic documents misclassifed by the CNN classifier and correctly classified by the BON classifier

Group	Num. doc.	Prop. of doc.	Avg. doc. length
BON misclassifed			
No named entities	57	0.31	27.14
Misleading named entities	17	0.09	38.18
Clarifying named entities	28	0.15	33.14
Unrelated named entities	83	0.45	35.04
All	185	1.00	32.61
CNN misclassifed			
No named entities	81	0.30	31.43
Misleading named entities	36	0.13	45.67
Clarifying named entities	58	0.21	35.53
Unrelated named entities	99	0.36	34.98
All	274	1.00	35.45

three versions of the system, the difference in performance is the smallest for the French language variety classification, only around one percentage point.

By combining both types of features, we manage to surpass the performance of the BON classifier on all language groups in the DSLCC v4.0 but the Malay and Indonesian pair. Here, the BON classifier beats the classifier with the combination of both types of features by a small margin of 0.2 percentage points. The synergy effect is the largest in case of Spanish language variety, where we improve the performance of the BON classifier by almost three percentage points. Overall performance of the classifier on all the languages is improved by about one percentage point in comparison to the BON classifier.

Results on smaller data sets are somewhat hard to generalize. In the case of ADIC, the performance gap between BON and CNN is almost 11 percentage points. The bad performance of the CNN classifier in this case also most likely outweighs any positive synergy effect, causing the classifier that uses a combination of both feature types to perform slightly (by about 0.3 percentage points) worse than the BON classifier (which is therefore a new state-of-the-art classifier for the ADIC data set). In the case of GDIC, the performance gap is smaller (about 1.3 percentage points) and there is some synergy effect between the two classifiers.

In order to determine what types of texts are better predicted with the BON classifier and what types of text are better predicted with the CNN classifier, we performed the same error analysis as in Section 5.3 on 185 Slavic documents, which were correctly classified by the CNN classifier and misclassified by the BON classifier, and on 274 documents which were correctly classified by the BON classifier and misclassified by the CNN classifier. Results are presented in Table 8. We can see that on average both of these documents are shorter (32.61 and 35.45 words long) than an average document in the Slavic sub-corpus (39.18 words long). Similar share of documents with no named entities was misclassified by both classifiers but there are differences in shares when it comes to other classes. Both BON and CNN classifiers performed the worst on documents containing only unrelated named entities but the share of these documents in the overall distribution of misclassified documents is much bigger for the BON classifier (0.45 vs. 0.36). On the other hand, documents containing clarifying named entities represent a smaller share in the distribution of documents misclassified by the BON classifier (0.15 vs. 0.21). These results are in accordance with the hypothesis that the BON classifier relies to a larger extent on named entities

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

than the CNN classifier. The share of documents with misleading named entities is the smallest in distributions for both classifiers, which was not the case in the error analysis in Section 5.3 (see Table 6), where the smallest share presented documents with only clarifying named entities. This suggests that both classifiers struggle with these documents and are in most cases misclassified by both classifiers; therefore (as this ablation study is focused on the differences between the BON and CNN classifiers), these documents were not manually analyzed.

7. Workflow for language variety classification

The AP—and larger NLP—community encourages reproducibility of results and code sharing[l]; therefore, our source code is published at `http://source.ijs.si/mmartinc/NLE_2017/`. Since AP is also a very interdisciplinary field, we also believe it is important to make our tools available to the users outside of the programming community (e.g., linguists or social scientists) with lower level of technical skills.

In our previous work (Martinc and Pollak 2018), we have already implemented a set of pretrained gender classification models into a cloud-based visual programming platform ClowdFlows (`http://clowdflows.org`) (Kranjc *et al.* 2012). These tools can be used out-of-the-box and are therefore appropriate for the less tech savy members of the AP community. The ClowdFlows platform employs a visual programming paradigm in order to simplify the representation of complex data mining procedures into visual arrangements of their building blocks. Its graphical user interface is designed to enable the users to connect processing components (i.e., widgets) into executable pipelines (i.e., workflows) on a design canvas, reducing the complexity of composition and execution of these workflows. The platform also enables online sharing of the composed workflows.

We took all our pretrained models for language variety classification (six models for six language groups and the general model for distinguishing between different language groups from the DSLCC v4.0, and German and Arabic models used for ADIC and GDIC classification) and packed them in a widget *Language Variety Classifier*. The widget takes a Pandas dataframe (McKinney 2011) containing the corpus as an input and returns a dataframe with an additional column with predicted language/language variety labels. The user needs to define the name of the column containing text documents as a parameter and choose the language group (or language parameter value *all* in order to use the general classifier) according to the input text.

Workflow in Figure 20 (available at `http://clowdflows.org/workflow/13322/`) is a ClowdFlows implementation of the two-step approach described in Section 4.2 for the language variety classification, illustrated on the DSLCC v4.0 test set. The corpus is loaded from a CSV file with two columns (one for texts and one for true labels) with the help of the *Load corpus from CSV* widget and passed on to the *Language variety classifier* widget, which predicts general language groups for all the texts. The *Filter corpus* widgets are used to split the corpus according to the predicted language group labels. Each of the slices is then fed into six different *Language variety classifier* widgets responsible for intra-language group classification. They output a Pandas dataframe with an additional column containing the predicted variety labels for each corpus slice. The corpus is reassembled with the help of the *Concatenate corpora* widget. The reassembled corpus and the six sub-corpora are then fed into seven *Calculate F1 and accuracy* widgets, which are in fact subprocess widgets[m], each of them containing a subprocess for calculating the accuracy and weighted F1 score[n] of the classification. The results of the classification are written to a table with the help of an *Evaluation results to table* widget. We have presented a repeatable and

[l]For example, this is the Github repository for the PAN shared task: `https://github.com/pan-webis-de`

[m]More information about the different types of widgets in the ClowdFlows platform is available at the ClowdFlows documentation page `https://clowdflows.readthedocs.io/en/latest/`.

[n]The results produced by the workflow vary very slightly from the results reported in Section 4 because Theano (Bergstra *et al.* 2011) is used as Keras backend in the ClowdFlows platform instead of Tensorflow (Abadi *et al.* 2016), which is used for producing the results reported in Section 4.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

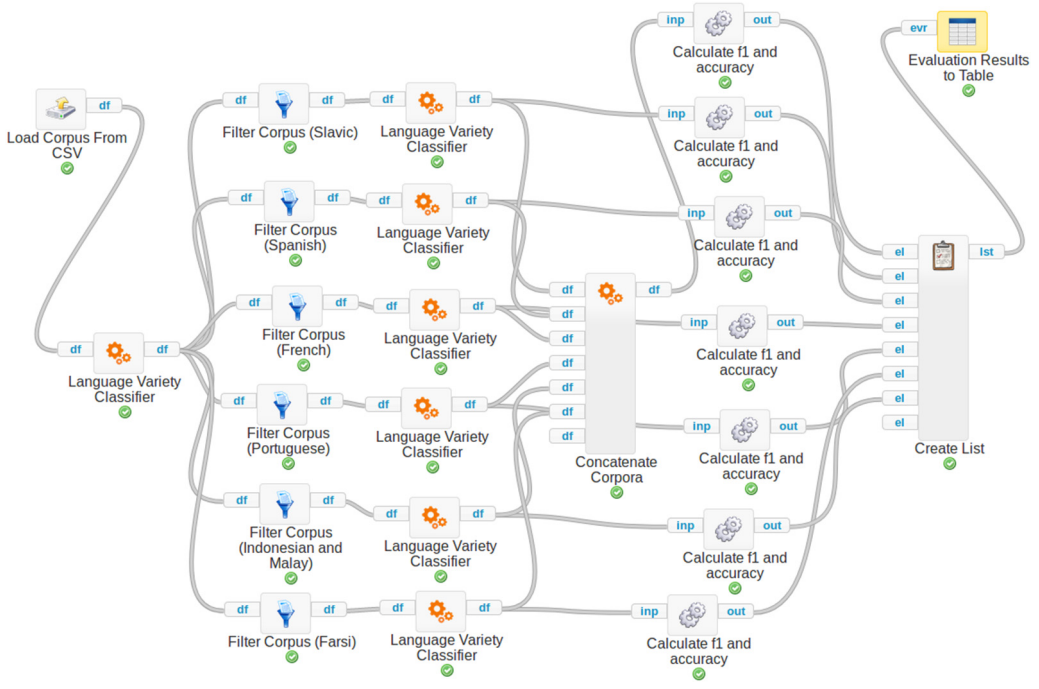

Figure 20. ClowdFlows implementation of the two-step approach for the language variety classification on the DSLCC v4.0. Workflow is publicly available at http://clowdflows.org/workflow/13322/.

transparent evaluation workflow, which can be easily tested on novel test sets, but note that the Language variety classifier widget can also be used in novel workflows, for assigning the language of unlabeled text segments. The simplest use would be to input a file with text that user wants to label in a CSV format and connect it to the two-step language classification widgets in order to obtain the labeled corpus (http://clowdflows.org/workflow/13670/).

8. Discussion and conclusions

In this paper, we present an original neural language variety classifier. The main novelty is the architecture that is capable of leveraging character-level and more global document/corpus-level information by combining weighted BON features with character-based CNN features. The system was tested on the DSLCC v4.0, ADIC and GDIC corpora, used in the VarDial shared tasks, and managed to outperform state-of-the-art approaches developed in the scope of the shared task on two (including on the benchmark DSLCC v4.0) out of three corpora. An ablation study shows that weighted BON features generally contribute more than CNN-based features. This is in accordance with the previous results in the AP shared tasks where BON-based classification systems were always the winners. On the other hand, our experiments showed that replacing TF-IDF weighting with BM25 weighting in most cases does not improve performance, which is not in accordance with the previous research (Bestgen 2017). Our system is also openly available as a workflow in the ClowdFlows platform for less tech savy members of the AP community.

The experiments on the DSLCC v4.0 have shown that building a neural architecture outperforming the popular SVM BON classification combination on the language variety task is possible, although the performance gains are not very large. With some additional language-group-specific parameter tweaking the performance could be improved, but we decided against this idea in order to preserve the generic nature of the common architecture, which is currently capable of producing state-of-the-art predictions for six different language groups.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

The system also proved to be competitive on the much smaller ADIC corpus (minimally outperforming state of the art) but failed to achieve competitive performance on GDIC (where the winning system HeLI was proposed by Jauhiainen *et al*. (2018a)).

We can speculate why this is the case. The results of the error analysis indicate a deterioration in performance of the proposed system on shorter documents. On the other hand, results of the VarDial 2018 shared tasks suggest that the performance of the HeLI system deteriorates less on shorter texts in comparison to other systems participating in shared tasks, since it ranked first on GDIC, where the documents are on average nine words long, and in the Vardial 2018 ILI shared task, where the task was to classify sentences[o], but only ranked fifth in the VarDial 2018 Discriminating between Dutch and Flemish in Subtitles task where the average document was 34.64 words long. Another hypothesis is that the proposed system is more reliant on named entities than the HeLI system, and therefore performs worse on GDIC, since this is the only corpus that does not contain news excerpts or news channel transcripts but transcripts of interviews with the dialect speakers and supposedly contains less named entities. We plan to test these hypotheses in the future work. We might also be able to boost the performance of our system on the GDIC data set by adjusting hyperparameters in order to make the network better suited for the classification of much shorter documents in the GDIC corpus, since currently a lot of data (e.g., *n*-grams that appear in less than five documents, character sequences filtered out by an aggressive max pooling ...) is discarded.

Small performance gains over the current state of the art also raise a question, how much better can automatic discrimination between similar languages actually get? The only study about the theoretical limit of the classification performance on the DSLCC that we are aware off was conducted by Goutte *et al*. (2016) on the DSLCC v2.0 used in the Vardial 2015 DSL shared task, which partially overlaps with the DSLCC v4.0 (Slavic, Malay and Indonesian, and Portuguese parts of the corpus are the same). First, they measured the upper bound on accuracy by taking all the predictions generated by all the systems which participated in the shared task and combining them using ensemble fusion methods such as plurality voting and Oracle. In the plurality voting, the label with most votes (i.e., the label predicted by most systems) is selected as correct and the conducted experiments showed that small improvements (of about 0.5 percentage point) over the best single system can be achieved. The Oracle method for determining the upper-bound performance on the other hand assigns the correct class label for an instance if at least one system classified the instance correctly. This gave them a very optimistic potential accuracy upper boundary of 99.83%.

In order to determine if the instances misclassified by the Oracle method can be correctly classified by humans, they conducted additional evaluation experiments. As it turns out, the difficulty of classification varies across different language groups. Discriminating between the three Slavic languages (Bosnian, Croatian, and Serbian) proved to be the most difficult. For 5 out of 12 instances misclassified by the Oracle method, none of the 6 annotators was able to correctly classify them. On these 12 examples the mean annotator accuracy was 16.66%, which is in fact 16.67% below the random baseline of 33.33%. On the other hand, discriminating between Brazilian and European Portuguese proved more feasible and the mean annotator accuracy on the misclassified instances was 67.50%, 17.50% above the 50% baseline.

This suggests that, at least for some varieties, the upper bound of automatic variety classification has not yet been reached, since our method achieved only 94.80% accuracy on the Portuguese language group. The conducted error analysis (see Section 5.3) on Slavic language varieties also showed that 10% of misclassified documents contained only clarifying named entities; therefore, any human annotator with some basic knowledge about Serbia, Bosnia, and Croatia would be able to classify them correctly without too much difficulty. This would suggest that further improvements on automatic language variety classification are possible, perhaps by employing

[o]We were unable to obtain the average document length for this data set since the number of tokens in the data set is not published in the Vardial 2018 report (Zampieri *et al*. 2018).

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

transfer learning techniques (Devlin *et al.* 2018) that would provide the classifier with the needed background information. We plan to test the transfer learning approach in the future.

CNNs have been so far the most successful neural architecture for language variety classification but the conducted ablation study shows that the produced features do have some deficiencies that make them less successful than weighted BON features. As shown, the proposed approach of feeding an additional weighted BON matrix into the network does partially compensate for these deficiencies on the language variety classification tasks but further work of exploring the synergy effects of combining automatically generated neural features and weighted features on a number of different NLP tasks and neural architectures is still needed. Feeding the sparse weighted BON matrix into the network does, however, have a drawback of drastically increasing the number of network parameters, which tends to lead to overfitting and increased computational costs. We managed to minimize these negative side effects mostly by an extensive use of dropout and by removing *n*-grams with low document frequencies from the input matrix, but perhaps a somewhat more efficient solution would be to avoid feeding the BON matrix to the neural classifier altogether. Therefore in future work, we plan to propose methods by which we would inject global document/corpus-level information into CNN-based features directly, in order to fix their current deficiencies. In that way combining them with the features that are the result of the more traditional feature engineering would no longer be required. Another option we also plan to explore is building heterogeneous ensembles of traditional SVM BOW-based models and CNNs and see if the performance gains are comparable to the proposed system.

Another line of future research will deal with building better and more useful tools for users with lower level of technical skills. Currently, the ClowdFlows platform does not support training of new neural classification models due to high level of resource consumption of these operations which would negatively affect the scalability of the platform, and since it does not yet support graphics processing unit (GPU) acceleration, which would allow for training of the models in a more reasonable time. The newer version of the ClowdFlows platform, on which the work has already begun, will address all these deficiencies and will allow for training of neural classification models on new varieties and therefore increase the overall usefulness of the system.

Acknowledgements. This paper is supported by European Union's Horizon 2020 research and innovation program under grant agreement No 825153—project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media) and grant agreement No 76966—project SAAM (Supporting Active Ageing through Multimodal coaching). The authors also acknowledge the financial support from the Slovenian Research Agency for research core funding for the program Knowledge Technologies (No P2-0103) and for the project TermFrame—Terminology and Knowledge Frames across Languages (No J6-9372). The results of this paper reflect only the author's view and the Commission is not responsible for any use that may be made of the information it contains. The Titan Xp used for this research was donated by the NVIDIA Corporation.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J. and Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In *OSDI,* vol. 16, pp. 265–283.

Ali, A., Dehak, N., Cardinal, P., Khurana, S., Yella, S.H., Glass, J. and Renals, S. (2015). Automatic dialect detection in arabic broadcast speech. In *Proceedings of Interspeech*, pp. 2934–2938. San Francisco, USA: ISCA.

Ali, M. (2018a). Character level convolutional neural network for German dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pp. 172–177. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Ali, M. (2018b). Character level convolutional neural network for Arabic dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pp. 122–127. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Ali, M. (2018c). Character level convolutional neural network for Indo-Aryan language identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pp. 283–287. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Alvarez-Carmona, M.A., López-Monroy, A.P., Montes-y-Gómez, M., Villasenor-Pineda, L. and Escalante, H.J. (2015). INAOE's participation at PAN'15: Author profiling task. In *Working Notes Papers of the CLEF*. Toulouse, France: CEUR Workshop Proceedings.

Belinkov, Y. and Glass, J. (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)*, pp. 145–152. Osaka, Japan: The COLING 2016 Organizing Committee.

Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H. and Nissim, M. (2017). N-gram: New Groningen author-profiling model. In *CLEF 2017 Evaluation Labs and Workshop - Working Notes Papers*. Dublin, Ireland: CEUR Workshop Proceedings.

Bergstra, J., Bastien, F., Breuleux, O., Lamblin, P., Pascanu, R., Delalleau, O. and Bengio, Y. (2011). Theano: Deep learning on GPUs with python. In *NIPS 2011, BigLearning Workshop, Granada, Spain*, vol. 3, pp. 1–48.

Bestgen, Y. (2017). Improving the character n-gram model for the DSL task with BM25 weighting and less frequently used feature sets. In *proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pp. 115–123. Valencia, Spain: Association for Computational Linguistics.

Bjerva, J. (2016). Byte-based language identification with deep convolutional networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pp. 119–125. Osaka, Japan: The COLING 2016 Organizing Committee.

Chollet, F. (2015). Keras: Deep learning library for theano and tensorflow. https://keras.io.

Cianflone, A. and Kosseim, L. (2017). 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)*, pp. 243–250. Osaka, Japan: The COLING 2016 Organizing Committee.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research* **12**, 2493–2537.

Çöltekin, Ç. and Rama, T. (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)*, pp. 15–24. Osaka, Japan: The COLING 2016 Organizing Committee.

Çöltekin, Ç., Rama, T. and Blaschke, V. (2018). Tübingen-Oslo team at the VarDial 2018 evaluation campaign: An analysis of n-gram features in language variety identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pp. 55–65. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Criscuolo, M. and Aluisio, S.M. (2017). Discriminating between similar languages with word-level convolutional neural networks. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pp. 124–130. Valencia, Spain: Association for Computational Linguistics.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Goutte, C., Léger, S. and Carpuat, M. (2014). The NRC system for discriminating similar languages. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pp. 139–145. Dublin, Ireland: Association for Computational Linguistics and Dublin City University.

Goutte, C., Léger, S., Malmasi, S. and Zampieri, M. (2016). Discriminating similar languages: Evaluations and explorations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 1800–1807. Portorož, Slovenia: European Language Resources Association.

Jauhiainen, T., Jauhiainen, H. and Lindén, K. (2018a). HeLI-based experiments in Swiss German dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pp. 254–262. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Jauhiainen, T., Jauhiainen, H. and Lindén, K. (2018b). Iterative language model adaptation for Indo-Aryan language identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pp. 66–75. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Joulin, A., Grave, E., Bojanowski, P. and Mikolov, T. (2016). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 427–431. Valencia, Spain: Association for Computational Linguistics.

Kingma, D.P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*. San Diego, California, USA: DBLP.

Kranjc, J., Podpečan, V. and Lavrač, N. (2012). ClowdFlows: A cloud based scientific workflow platform. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 816–819. Bristol, UK: Springer.

López-Monroy, A.P., Montes-y-Gómez, M., Escalante, H.J. and Pineda, L.V. (2014). Using intra-profile information for author profiling. In *CLEF (Working Notes)*, pp. 1116–1120. Sheffield, UK: CEUR Workshop Proceedings.

Malmasi, S. and Dras, M. (2015). Language identification using classifier ensembles. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, pp. 35–43. Hissar, Bulgaria: Association for Computational Linguistics.

https://doi.org/10.1017/S1351324919000299 Published online by Cambridge University Press

Malmasi, S. and Zampieri, M. (2016a). Arabic dialect identification in speech transcripts. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pp. 106–113. Osaka, Japan: The COLING 2016 Organizing Committee.

Malmasi, S., Zampieri, M., Ljubešić, N., Nakov, P., Ali, A. and Tiedemann, J. (2016b). Discriminating between similar languages and arabic dialect identification: A report on the third DSL shared task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pp. 1–14. Osaka, Japan: The COLING 2016 Organizing Committee.

Martinc, M. and Pollak, S. (2018). Reusable workflows for gender prediction. In *Language Resources and Evaluation Conference (LREC 2018) Proceedings*, pp. 515–520. Miyazaki, Japan: European Language Resources Association.

McKinney, W. (2011). Pandas: A foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, pp. 1–9.

Miura, Y., Taniguchi, T., Taniguchi, M. and Ohkuma, T. (2017). Author profiling with word + character neural attention network. In *CLEF (Working Notes)*. Dublin, Ireland: CEUR Workshop Proceedings.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. and Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research* **12**, 2825–2830.

Rangel, F., Rosso, P., Koppel, M., Stamatatos, E. and Inches, G. (2013). Overview of the author profiling task at PAN 2013. In *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pp. 352–365. Valencia, Spain: Springer.

Rangel Pardo, F.M., Celli, F., Rosso, P., Potthast, M., Stein, B. and Daelemans, W. (2015). Overview of the 3rd author profiling task at PAN 2015. In *CLEF 2015 Evaluation Labs and Workshop Working Notes Papers*, pp. 1–8. Toulouse, France: CEUR Workshop Proceedings.

Rangel, F., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M. and Stein, B. (2016). Overview of the 4th author profiling task at PAN 2016: Cross-genre evaluations. In Balog, K. et al. (ed.) *Working Notes Papers of the CLEF 2016 Evaluation Labs*. CEUR Workshop Proceedings, pp. 750–784. Évora, Portugal: CEUR Workshop Proceedings.

Rangel, F., Rosso, P., Potthast, M. and Stein, B. (2017). Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. In *Working Notes Papers of the CLEF*. Dublin, Ireland: CEUR Workshop Proceedings.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval* **3**(4), 333–389.

Samardzic, T., Scherrer, Y. and Glaser, E. (2016). Archimob-a corpus of spoken Swiss German. In *Proceedings of LREC 2016*, pp. 4061–4066. Portorož, Slovenia: European Language Resources Association.

Stamatatos, E., Daelemans, W., Verhoeven, B., Potthast, M., Stein, B., Juola, P. and Barrón-Cedeño, A. (2014). Overview of the author identification task at PAN 2014. In *CLEF 2014 Evaluation Labs and Workshop Working Notes Papers, Sheffield, UK, 2014*, pp. 1–21. Sheffield, UK: CEUR Workshop Proceedings.

Tan, L., Zampieri, M., Ljubešic, N. and Tiedemann, J. (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)*, pp. 11–15. Reykjavik, Iceland: European Language Resources Association.

Vollenbroek, M.B., Carlotto, T., Kreutz, T., Medvedeva, M., Pool, C., Bjerva, J. and Nissim, M. (2016). Gronup: Groningen user profiling. In *Notebook for PAN at CLEF*, pp. 846–857. Évora, Portugal: CEUR Workshop Proceedings.

Zampieri, M., Tan, L., Ljubešić, N. and Tiedemann, J. (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*, pp. 58–67. Dublin, Ireland: Association for Computational Linguistics and Dublin City University.

Zampieri, M., Tan, L., Ljubešić, N., Tiedemann, J. and Nakov, P. (2015). Overview of the DSL shared task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, pp. 1–9. Hissar, Bulgaria: Association for Computational Linguistics.

Zampieri, M., Malmasi, S., Ljubešić, N., Nakov, P., Ali, A., Tiedemann, J. and Aepli, N. (2017). Findings of the VarDial evaluation campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pp. 1–15. Valencia, Spain: Association for Computational Linguistics.

Zampieri, M., Malmasi, S., Nakov, P., Ali, A., Shon, S., Glass, J. and Van der Lee, C. (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 2018)*, pp. 1–17. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Cite this article: Martinc M and Pollak S. Combining *n*-grams and deep convolutional features for language variety classification. *Natural Language Engineering* **25**, 607–632. https://doi.org/10.1017/S1351324919000299