Comparing the Performance of CNNs and Shallow Models for Language Identification

Andrea Ceolin
Università di Modena e Reggio Emilia
ceolin@unimore.it

Abstract

In this work we compare the performance of convolutional neural networks and shallow models on three out of the four language identification shared tasks proposed in the VarDial Evaluation Campaign 2021. In our experiments, convolutional neural networks and shallow models yielded comparable performance in the Romanian Dialect Identification (RDI) and the Dravidian Language Identification (DLI) shared tasks, after the training data was augmented, while an ensemble of support vector machines and Naïve Bayes models was the best performing model in the Uralic Language Identification (ULI) task. While the deep learning models did not achieve state-of-the-art performance at the tasks and tended to overfit the data, the ensemble method was one of two methods that beat the existing baseline for the first track of the ULI shared task. [1]

1 Introduction

In this paper, we present the submissions of Team Phlyers to the VarDial Evaluation Campaign 2021 (Chakravarthi et al., 2021). The campaign is part of a conference series, the Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial), which has reached its eighth edition, five of which have included several shared tasks (Zampieri et al., 2017, 2018, 2019; Găman et al., 2020). The shared tasks typically involve the categorization of texts according to their language or their dialect. This task, known as language identification, is a classical NLP problem (House and Neuburg, 1977; Dunning, 1994; McNamee, 2005), because of its importance for information retrieval, machine translation, and more recently categorization of social media posts (Bergsma et al., 2012; Lui and Baldwin, 2014; Zubiaga et al., 2016).

In the next sections, we briefly describe the three tasks that we participated in.

1.1 RDI

The VarDial 2019 edition (Zampieri et al., 2019) proposed the first shared task based on distinguishing standard Romanian from Moldavian newspaper articles. The task consisted in training a classifier on news articles in Romanian and Moldavian from the MOROCO corpus (Butnaru and Ionescu, 2019), and using it to classify other news articles yet to be added to the corpus. The best model achieved an F_1 score of 0.895 on the test set, using an ensemble method based on convolutional neural networks (CNN) and support vector machines (SVM) (Tudoreanu, 2019).

Last year's task asked participants to train a classifier on the news articles of the MOROCO corpus to distinguish standard Romanian from Moldavian tweets (Găman et al., 2020). The task was particularly interesting because the organizers provided a large validation dataset based on news articles (5923), and only a small validation dataset based on tweets (215). This made the validation stage challenging, because on the one hand a model trained on news articles which yields high accuracy on the news validation dataset could fail to generalize to a different domain, while on the other hand the size of the tweets validation dataset was so small that by training a model only on tweets, the risk of overfitting was considerable. The best result on the task was obtained by an ensemble of linear SVM classifiers (Çöltekin, 2020), which yielded an accuracy of F_1=0.788 on the test data. A similar accuracy (F_1=0.775) was reached by fine-tuned Romanian BERT models (Popa and Stefănescu, 2020). Attempts that relied on Naïve Bayes (NB) models yielded lower accuracies (Jauhiainen et al., 2020a; Ceolin and Zhang, 2020).

The shared task has been proposed again in this

[1] The code for the models employed in this work is found at https://github.com/AndreaCeolin/VarDial2021.

Proceedings of the 8th VarDial Workshop on NLP for Similar Languages, Varieties and Dialects, pages 102–112
April 20, 2021 ©2021 Association for Computational Linguistics

year's VarDial edition, in which a new dataset of tweets in standard Romanian and Moldavian was made available to the participants.

1.2 ULI

The Uralic Language Identification task (ULI) differs from traditional tasks in the high number of language varieties that are present in the dataset (Jauhiainen et al., 2020b; Găman et al., 2020). In total, there are 178 languages that need to be distinguished, which include the 29 Uralic varieties which are the focus of the task, 3 extra Uralic languages (Finnish, Estonian, and Hungarian) and 146 other non-Uralic languages. Moreover, the classes are imbalanced: for the 29 Uralic varieties, the number of sentences vary from 19 to 214,225, while for the other languages the range is much higher (from 10K to 3 million).

The shared task is divided in three separate subtasks. The first subtask requires a model to distinguish among the 29 Uralic varieties giving equal importance to each of the classes, and is evaluated through a macro F_1 score. In the second subtask, classes are weighted according to their frequency, and thus a micro F_1 score is used. In the third subtask, the models are evaluated on all 178 languages.

The high number of classes, their overlap, and the imbalanced dataset all represent challenges for deep learning algorithms: in particular, they can lead to overfitting if the traning set is not balanced (Bernier-Colborne and Goutte, 2020). On the other hand, the same problem might affect shallow methods like SVMs, since they need to identify many distinct separation hyperplanes in a domain where many languages are virtually indistinguishable.

The shared task was presented for the first time during the VarDial Evaluation Campaign 2020 (Găman et al., 2020), and in that case the best performing system for all tracks was the HeLI method (Jauhiainen et al., 2016), which was the baseline presented by the organizers. Since the HeLI baselines were not improved, the task has been proposed again in this year's edition.

1.3 DLI

The Dravidian Language Identification task (DLI) requires participants to classify three South Dravidian languages (Tamil, Malayalam, and Kannada) using a dataset of 16,672 YouTube comments (Chakravarthi et al., 2020b,a; Hande et al., 2020). The comments are written in Roman script.

This task differs from the others in being focused on code-switching: all comments contain a mix of words from the target language and words from English, and in some cases native words can appear within an English grammatical structure. Classifiers must then be robust to this variability and be able to not be deceived by the English material. Another interesting feature of this task is the large class imbalance in the training set (with about 10K comments in Tamil, 4K in Malayalam, and only about 500 in Kannada) and the fact that both the training and the test datasets contain comments from other languages (under the label 'other-language', approximately 1K comments).

This is the first edition of this task.

2 Methods

Previous methods used for language identification typically involve SVMs (Goutte et al., 2014; Çöltekin and Rama, 2017; Medvedeva et al., 2017; Kreutz and Daelemans, 2018; Benites de Azevedo e Souza et al., 2018; Wu et al., 2019) and multinomial NB models applied to word and character ngrams (Barbaresi, 2016; Clematide and Makarov, 2017; Jauhiainen et al., 2016, 2020a). Deep learning methods based on CNNs and LSTMs have also been successfully applied to language identification tasks (Jaech et al., 2016; Butnaru and Ionescu, 2019; Hu et al., 2019; Tudoreanu, 2019), and the last two editions of VarDial also showed successful applications of BERT models (Bernier-Colborne et al., 2019; Popa and Stefănescu, 2020; Scherrer and Ljubešić, 2020; Zaharia et al., 2020).

For the current tasks, we employed NB and SVM models trained on character ngrams as baselines, and compared their performance with that of a CNN. For the CNN, we decided to use the character-based model for text classification that was developed by Zhang et al. (2015), and that was successfully adopted by Butnaru and Ionescu (2019) to distinguish between standard Romanian and Moldavian news texts. The architecture of the CNN is summarized in Table 1. All models were run using Google Colab, with 1 GPU.[2]

3 Results

This section summarizes our contributions to the three shared tasks, the evaluation of our models, and their performance on the test datasets.

[2] https://colab.research.google.com

	Type	In	Out	Kernel	MaxPool	Stride
1	Conv.	(*)	128	7	3	3
2	Conv.	128	128	7	3	3
3	Conv.	128	128	3	3	3
4	Linear	(*)	1000	-	-	-
5	Linear	1000	1000	-	-	-
6	Linear	1000	(*)	-	-	-

Table 1: CNN architecture. Filters applied over a single dimension. Loss function=CrossEntropy, Optimizer = Adam. DropOut is added to the two fully connected layers (0.5). ReLu threshold: 10^{-6}. Output is passed through Softmax. The dimensions marked with (*) are task specific.

3.1 RDI

A question that remains open after last year's edition is whether deep learning methods can achieve a good accuracy at distinguishing between standard Romanian and Moldavian tweets even with limited training data. The deep learning models that were used in last year's task (Popa and Stefănescu, 2020; Zaharia et al., 2020) all relied on pre-trained BERT models (Dumitrescu et al., 2020), which might not always be available when working on low-resource languages. Even though CNNs have been successfully applied to the task of distinguishing news texts between the two language varieties (Butnaru and Ionescu, 2019), in our contribution to last year's edition we have showed that the representations they learn fail to generalize to the tweets domain (Ceolin and Zhang, 2020). This year we readdressed this issue using the new dataset.

3.1.1 CNN

In last year's edition, a small dataset of 215 tweets was given to the participants to evaluate the model. Since this year's edition provides a larger amount of in-domain data (5237 tweets), we decided to use the model we trained last year and fine-tune it using the larger tweets validation dataset available for this year's task. We also decided to train a separate model on tweets-data only, to see if the use of out-of-domain news data leads to a better performance than a model trained only on tweets.

In order to augment the data, we experimented with some of the data augmentation techniques proposed by Wei and Zou (2019). The one which turned out to be the most successful was random swap, especially when it was used multiple times on the same sentence rather than just once (i.e., essentially shuffling the words in the sentence). See the Appendix for a more detailed summary of the data augmentation experiments.

After some trial runs, we decided to set a batch size of 128, and a learning rate of 0.001. We used 1/5 of the data to create a validation dataset, while the rest was used for training. The training data is augmented with 10 replications that involve shuffled sentences. On the basis of training and validation accuracies, we decided to interrupted training after 10 epochs on the original training data, and after 5 epochs on the augmented dataset.

3.1.2 Shallow models

We also trained a NB and a linear SVM model on TFIDF-transformed character ngrams in the [5-8] range, which was determined to be the optimal range for these languages in Ceolin and Zhang (2020). The models have not been fine-tuned, and have been evaluated using the same validation dataset selected to evaluate the CNN.

3.1.3 Evaluation

The tweets dataset was already balanced, with 2625 standard Romanian tweets and 2612 Moldavian tweets. As we see in Table 2, data augmentation improved the performance of the CNNs dramatically, to the point that it became comparable to that of shallow models.

Model	Macro F_1 score
NB	0.760
CNN (news+tweets) + data aug.	**0.756**
Linear SVM	0.756
CNN (tweets) + data aug.	0.749
CNN (news+tweets)	0.709
CNN (tweets)	0.700

Table 2: Final performance of the models on the evaluation of the RDI task.

For instance, if we take a look at the CNN (news+tweets) model, Figure 1 shows that after training for 10 epochs the performance on the validation set reaches a macro F_1 score of 0.709, while in Figure 2 we see that in the augmented dataset the accuracy is well above 0.7 after the first epoch, and converges to ≈0.76 after five epochs. The same is true for the model trained only on tweets, whose accuracy jumps from 0.7 to 0.75.

3.1.4 Results

We decided to submit two runs to the RDI shared task. Both contained the predictions of the CNN pre-trained on news articles and then fine-tuned on augmented tweets, but the second submission

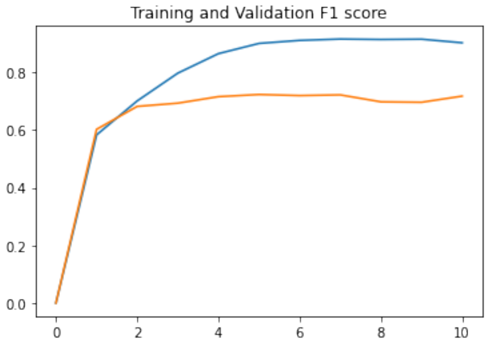

Figure 1: CNN news + tweets model. Comparing the training and validation performance measured by macro F_1 score through 10 epochs of training. Blue line: training; orange line: validation. The training dataset is not augmented.

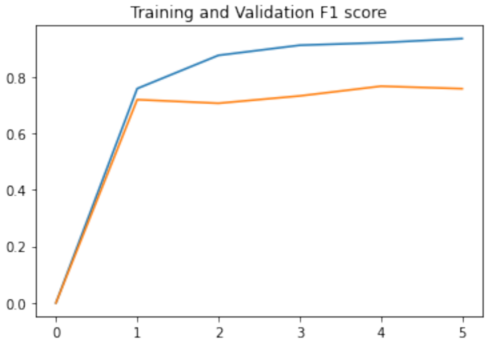

Figure 2: CNN news + tweets model. Comparing the training and validation performance measured by macro F_1 score through 5 epochs of training. Blue line: training; orange line: validation. Training set augmented.

resulted from a model were data augmentation had an increased weight (from x10 to x12).

Table 3 contains the results of the best runs of the three teams that took part in the task. The best accuracy (0.777) was reached by SUKI, who last year proposed a NB model trained on character ngrams (Jauhiainen et al., 2020a), while the second best accuracy (0.732) was reached by the UPB team (Zaharia et al., 2020), who last year addressed the task using a BERT model. The CNN we presented did not reach comparable performance, even though an inspection of precision and recall did not point to any obvious explanation: the network just ended up overfitting the training data. Our second run was not competitive, because increased data augmentation without changing the other parameters of the CNN led to even more overfitting.

Model	Macro F_1 score
SUKI	0.777
UPB	0.732
Phlyers	**0.653**

Table 3: Final performance of the teams' submissions to the RDI task.

Considering that the network was trained on the full tweets dataset rather than 80% of it before the submission, stopping the iterations after the first or the second epoch would have been a wiser choice, since we have seen that no considerable improvement of the valuation performance was made after the first few epochs, and therefore additional training might have led to overfitting.

3.2 ULI

As we previously mentioned, this task appeared to be the most challenging one, given the high amount of labels and the great class imbalance. In particular, the first subtask required the models to be able to accurately classify languages which were represented by a few dozen sentences.

The strategy we adopted to address these issues was to train two separate classifiers for two different classification problems. First, we want to be able to distinguish the 29 'target' Uralic languages from the 149 'non-target' languages. Once we have a model that can distinguish the two types of languages, we can train a second classifier to distinguish among the 29 target languages using features only extracted from such languages. In this way, the second model would be able to extract more features which are only needed to separate the 29 target languages, even though employing them in the first stage might produce some errors, in addition to more computational cost.

3.2.1 Distinguishing 'target' and 'non-target' languages

Defining a unique alphabet for the entire training dataset is unfeasible, since languages make use of different writing systems, and this fact makes it practically difficult to train a character-based CNN for the task. Therefore, we decided to focus on shallow models in this first stage. For this binary classification task, we trained a linear SVM and a NB method. In addition to the 646,043 sentences for the target languages, we used 5000 sentences for each of the non-target language, obtaining a total of 1,391,043 sentences for the 178 languages of the task. We then used 5-fold cross-validation to evaluate the classifier.

After some trial runs, we found that a linear SVM classifier trained on TF-IDF transformed character ngrams in the range [3,4], with the number of features limited to the 100,000 most frequent ones, converges to a 0.995 macro F_1 score. We will then use this classifier to single out the target languages in the test dataset.

Model	Ngrams	Macro F_1 score
Linear SVM	[3]	0.992
Linear SVM	[4]	0.994
Linear SVM	[5]	0.991
Linear SVM	[3,4]	**0.995**
NB	[3]	0.976
NB	[4]	0.986
NB	[5]	0.988
NB	[4,5]	0.985

Table 4: Evaluation of the models used to distinguish 'target' from 'non-target' languages.

3.2.2 Classifying 'target' languages

We first attempted to train a CNN for this subtask, but we could not develop a system that was able to deal with the large class imbalance and with the high number of different labels that need to be learned. We then retrained our shallow models for the task of distinguishing among the target languages. The best performing model was a NB model trained on TFIDF character ngrams, with ngrams in the range [3,5] and alpha=10^{-6} (cf. Table 5). Rare ngrams, those whose relative document frequency was less than $<10^{-4}$, were excluded. The model parameters were selected through 5-fold cross-validation.

Model	Ngrams	Macro F_1	Micro F_1
Linear SVM	[3]	0.748	0.964
Linear SVM	[4]	0.765	0.965
Linear SVM	[5]	0.772	0.962
NB	[3]	0.871	0.976
NB	[4]	0.909	0.983
NB	[5]	0.911	0.985
NB	[4,5]	0.907	0.985
NB	[3,5]	**0.915**	**0.986**

Table 5: Evaluation of the models used to distinguish among the 'target' languages.

3.2.3 Evaluation - Track 1 and 2

In order to evaluate our system, we decided to create a 80/20 training-test split. First, the SVM is trained on 80% of the dataset in order to determine whether the sentences in the test set are from target or non-target languages. Then, the NB model assigns a label to the sentences that are recognized as target sentences. The results are in Table 6.

Model	Macro F_1	Micro F_1
Linear SVM + NB	**0.905**	**0.988**

Table 6: Evaluation of the final model.

We see that the full model yields a macro F_1 score of 0.905. We noted that precision was higher than recall (0.958 versus 0.878), and in particular rare languages are associated to low recall scores. This means that the system does not make 'wrong' predictions often, but it can fail to identify the less common varieties.

The micro F_1 score for the system is about 0.99, as expected.

3.2.4 Evaluation - Track 3

The model we devised could not be used for a submission to Track 3, because the non-target languages are excluded in the first step. For this reason, we decided to retrain our models to directly predict the labels, and to use 5-fold cross-validation to evaluate them.

In this case, we need to increase the threshold for filtering rare ngrams to 10^{-3} for memory constraints, and to limit the analysis to NB systems, since the high number of labels makes working with SVMs more challenging. The best model we selected was a NB model with TFIDF transformed character 5-grams (F_1=0.949, see Table 7).

Model	Ngrams	Macro F1
NB	[3]	0.937
NB	[4]	0.946
NB	[5]	**0.949**

Table 7: Evaluation of the models developed for Track 3.

3.2.5 Results

We submitted the two systems we described in the preceding sections (SVM+NB for Track 1 and 2, and NB for Track 3) for evaluation. In addition, we submitted the prediction of two ensemble models, that we summarize in Table 8. The models are derived from the two main models, but their predictions change when the two main models are in disagreement about which of two target languages is the correct label. The first ensemble (Ensemble$_1$) model uses all the predictions of the model developed for Track 1 and 2, but there is one case in which the prediction of the model developed for Track 3 are selected instead: when the first model predicts one of the five rare languages of the dataset, Ingrian (izh), Nganasan (nio), Kemi Sami (sjk), Ume Sami (sju), Votic (vot), and the second model predicts a different target language, then this second target language is selected instead. Our motivation is the following: if the model erroneously predicts a language which is not in the test set, precision for that language will go to zero, and the performance of the classifier will significantly drop. This strategy will make sure that the prediction is indeed accurate when dealing with rare languages. In this case, if the signal between the two classifiers is conflicting, it appears wiser to default to the prediction of the most common class.

The second ensemble (Ensemble$_2$) takes the predictions of the NB model developed for Track 3, but when the SVM+NB model predicts a different target language, then this more refined prediction is chosen instead, under the assumption that since the SVM+NB model has been developed specifically for distinguishing among target languages, its predictions should be more accurate.

The results for the three tasks are summarized in Table 9, Table 10 and Table 11.

As for Track 1 (Table 9), the first ensemble system was the best performing system among those we submitted, and the second best overall, although all teams provided systems with similar performances and architectures. It is noticeable how combining the predictions of the two different systems we devised for the task, with the aim of improving the classification of the rare languages, led to a significant improvement over the performance of the two systems taken separately.

SVM+NB (1,2)	NB (3)	Ensemble$_1$	Ensemble$_2$
NA	NA	NA	NA
NA	Target_b	NA	Target_b
Target_a	NA	Target_a	NA
Target_a	Target_b	$\mathbf{Target_{a\text{-}b}}$	$\mathbf{Target_a}$

Table 8: Ensemble methods developed for the three subtasks. Target_a refers to the prediction of a language on target made by the classifier developed for Track 1 and 2, while Target_b refers to the prediction of a language on target made by the classifier developed for Track 3. Note how the predictions of the two ensemble models are essentially the same, respectively, of the two main models, but they differ in how they assign a label to a target language when there is disagreement between the two main classifiers.

Team	Model	Macro F$_1$
NRC	Probabilistic Classifier, ch.5grams	0.8138
Phlyers	**Ensemble$_1$ (SVM+NB), ch. 3-5grams**	**0.8085**
Phlyers	Ensemble$_2$ (SVM+NB), ch. 3-5grams	0.8076
SUKI	HeLI	0.8004
Phlyers	NB ch. 5grams	0.7977
LAST	Logistic R., ch.1-3grams, BM25	0.7977
Phlyers	SVM (ch.3-4grams)+NB (ch.3-5grams)	0.7740

Table 9: Results for Track 1.

As for Track 2 (Table 10), Ensemble$_1$ and the SVM+NB model yielded the same performance (0.84), which was clearly below the baseline established by HeLI (Jauhiainen et al., 2020b), and below the performance of the systems submitted by the other teams. During the evaluation of our submissions, the organizers also provided us with the precision and recall scores, and it was clear that the failure was entirely due to the low precision of the systems. Since our evaluation set was balanced between target and non-target languages (with about 30% of the sentences belonging to the target set), the precision scores looked acceptable, but an error analysis clearly showed that the systems had the tendency of assigning a target label to a non-target language more often than the opposite, even though it was precisely this behavior that we were hoping to avoid with the SVM classifier.

Since our system still failed to filter out some non-target languages, precision was drastically reduced in the test phase, where non-target languages clearly outnumbered target languages (sentences belonging to target languages were about 2% of the whole sample, according to the SVM model).

Indeed, we also submitted the model we developed for Track 3, and the second ensemble, but both attempts yielded an even lower performance, which suggests that the SVM filter was partially successful at filtering out non-target languages, even though it was not sufficient to achieve state-of-the-art performance.

Team	Model	Micro F_1
NRC	Probabilistic Classifier using ch. 5grams	0.9668
SUKI	HeLI	0.9632
NRC	BERT Deep Neural Network	0.9530
LAST	Logistic R., ch.1-3grams, BM25	0.9496
Phlyers	**SVM(ch.3-4grams)+NB(ch.3-5grams)**	**0.8389**
Phlyers	NB ch. 5grams	0.5934
Phlyers	Ensemble$_2$ (SVM+NB), cf.3-5grams	0.5932

Table 10: Results for Track 2.

Finally, for Track 3 (Table 11) none of the systems submitted to the task were able to beat the strong baseline set by HeLI. Even though in this case the performance of our systems was close to that of the systems submitted by LAST and NRC, ours were not able to reach the performance of the other teams.

Team	Model	Macro F_1
SUKI	HeLI	0.9252
LAST	Logistic R., ch.1-3grams, BM25	0.9164
NRC	Probabilistic Classifier, ch.5grams	0.9079
NRC	BERT Deep Neural Network	0.9039
Phlyers	**Ensemble$_2$ (SVM+NB), ch.3-5grams**	**0.8847**
Phlyers	NB ch. 5grams	0.8831

Table 11: Results for Track 3.

3.3 DLI

As we said above, this task is focused on code-switching, and this means that many features that could be extracted are completely irrelevant to determine the original language of the text. The great class imbalance is another problem that needs to be addressed. Especially in the design of deep learning architectures, some strategy to prevent overfitting was required.

3.3.1 CNN

We addressed this task using the same CNN developed for the RDI task, with an important difference. Since in this case we have to deal with class imbalance, we decided to perform balanced sampling during the training phase.

First, 1/5 of the labeled data was randomly selected for evaluation purposes as a validation dataset, and the rest was used for training. Then, we sampled a total of 25,000 sentences uniformly across the four categories by selecting each of the four classes with p=0.25, and each sentence with p=1/n_C, with n_C being the number of sentences available for each class. This will necessarily imply that many sentences will be picked more than once, especially for the classes which are not well represented.

In order to avoid repeating the same sentences for the more uncommon classes, we decided to shuffle the order of the words in the sentences, essentially adopting the data augmentation strategy that was employed for the RDI task. This strategy had two purposes: dealing with class imbalance by augmenting the data of the classes which were not well represented, and addressing the problem of the influence of the English grammar, by exposing the network to sentences in which the order of the words was changed, with the aim of retrieving word sequences that were not in the training data, but were still possible in the language. We also trained a separate model where instead the order of the words was not shuffled, and therefore sentences in the training dataset were just repeated.

Since most of the comments are short, only the first 160 characters per comment were used as input to the network. After some parameter tuning, we set the learning rate to 0.001, and the batch size is 256. We also reduced the output of the second linear layer to 500. Training was interrupted after 10 epochs.

3.3.2 Shallow models

Following the strategy adopted for the RDI task, we trained a NB and a linear SVM model on TFIDF-transformed character ngrams in the [5-8] range. The models have not been fine-tuned, and have been evaluated using the same dataset used to evaluate the CNN.

3.3.3 Evaluation

Table 12 shows the micro F_1 score, which was the metric used to rank the submissions, for the models evaluated.

The patterns are similar to those we have obtained in the RDI task: shuffling words had the effect of improving the performance of the CNN. Figure 3 and Figure 4 show that in this case shuffling had only a marginal effect on the task, since in both cases training and validation performances were comparable.

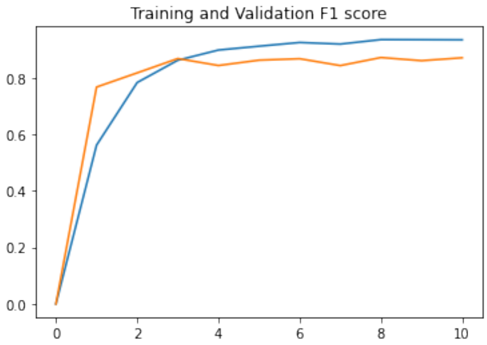

Figure 3: CNN model. Comparing the training and validation performance measured by micro F_1 score through 10 epochs of training. Blue line: training; orange line: validation. Sentences are not shuffled.

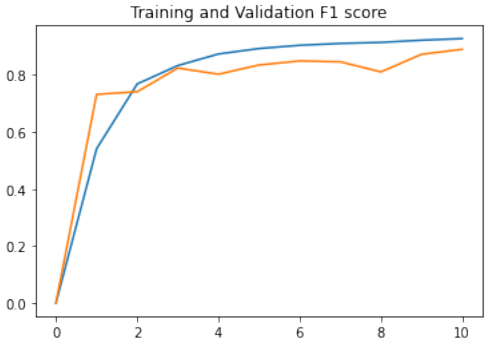

Figure 4: CNN model. Comparing the training and validation performance measured by micro F_1 score through 10 epochs of training. Blue line: training; orange line: validation. Sentences are shuffled.

We then submitted our best model for evaluation.

Model	Micro F_1 score
CNN (shuffle)	**0.880**
NB	0.878
CNN (no shuffle)	0.870
Linear SVM	0.848

Table 12: Final performance of the models on the evaluation of the DLI task.

3.3.4 Results

The results of the evaluation campaign are in Table 13. Our model performed as expected, with a micro F_1 score of 0.9. The performance was comparable to the submission of the other three teams, which however all yielded a better F_1 score.

The only systematic difference between our submission and the others was a low F_1 score for the 'other-languages' class (0.46), while all teams were able to achieve a score of at least 0.54. This suggests that our network was not able to obtain a representation of this class as robust as that obtained by the other classifiers.

Model	Micro F_1 score
LAST	0.93
Nayel	0.92
HWR	0.92
Phlyers	**0.90**

Table 13: Final performance of the teams' submissions to the DLI task.

4 Conclusion

The last editions of the VarDial evaluation campaign (Zampieri et al., 2019; Găman et al., 2020) have seen an increased use of deep learning techniques for language identification, which in several cases yielded the best performance at the tasks (Tudoreanu, 2019; Bernier-Colborne et al., 2019). In this work, we tried to compare the performance of CNNs and shallow models for three out of the four tasks at VarDial 2021. While for the ULI task developing a CNN turned out to be challenging, for both the RDI and the DLI task CNNs yielded performances which were in line with the baselines established by the more classic shallow models, even though the final results showed that they are prone to overfitting.

It is interesting to note that shuffling the words in the training data improved the accuracy of our CNN classifiers, in particular for the RDI task. The procedure essentially introduces noise in the data, because the order of the words in the sentences will be ungrammatical after they are shuffled, so why it improves the performance of the classifier is not clear. One possibility is that it introduces the network to word combinations that would be possible in the language (for instance, but switching a subject and an object, or by juxtaposing words separated by modifiers), increasing the diversification of the training data. Another possibility is that since shuffling words does not affect character sequences within words, but at word boundaries, shuffling has the effect of preventing the network from focusing on sequences with spaces in the middle, which could be less meaningful than sequences within words to learn the lexicon and the morphology associated to each language variety. In the current experiments, this strategy had the effect of reducing overfitting. This outcome will require more investigation in the future.

While data augmentation is popular in image classification (Wang and Perez, 2017; Cubuk et al., 2019), it has so far had limited application in NLP (Coulombe, 2018; Kobayashi, 2018; Wei and Zou, 2019). Our experiments on the VarDial 2021 shared tasks suggest that data augmentation can play an important role in adapting neural models to the task of language identification.

Acknowledgments

Thanks to the co-founder of Team Phlyers, Hong Zhang.

References

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.

Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. 2012. Language identification for creating language-specific Twitter collections. In *Proceedings of the second workshop on language in social media*, pages 65–74.

Gabriel Bernier-Colborne and Cyril Goutte. 2020. Challenges in neural language identification: NRC at VarDial 2020. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 273–282.

Gabriel Bernier-Colborne, Cyril Goutte, and Serge Léger. 2019. Improving cuneiform language identification with BERT. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 17–25.

Andrei M. Butnaru and Radu Tudor Ionescu. 2019. MOROCO: The Moldavian and Romanian Dialectal Corpus. In *Proceedings of ACL*, pages 688–698.

Andrea Ceolin and Hong Zhang. 2020. Discriminating between standard Romanian and Moldavian tweets using filtered character ngrams. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 265–272.

Bharathi Raja Chakravarthi, Mihaela Găman, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubešić, Niko Partanen, Ruba Priyadharshini, Christoph Purschke, Eswari Rajagopal, Yves Scherrer, and Marcos Zampieri. 2021. Findings of the VarDial Evaluation Campaign 2021. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020a. A sentiment analysis dataset for code-mixed Malayalam-English. pages 177–184, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020b. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210, Marseille, France. European Language Resources association.

Simon Clematide and Peter Makarov. 2017. CLUZH at VarDial GDI 2017: Testing a variety of machine learning tools for the classification of swiss German dialects. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 170–177, Valencia, Spain. Association for Computational Linguistics.

Çağrı Çöltekin. 2020. Dialect identification under domain shift: Experiments with discriminating Romanian and Moldavian. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 186–192.

Çağrı Çöltekin and Taraka Rama. 2017. Tübingen system in VarDial 2017 shared task: experiments with language identification and cross-lingual parsing. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 146–155, Valencia, Spain.

Claude Coulombe. 2018. Text data augmentation made simple by leveraging NLP Cloud APIs. *arXiv preprint arXiv:1812.04718*.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123.

Stefan Dumitrescu, Andrei-Marius Avram, and Sampo Pyysalo. 2020. The birth of Romanian BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4324–4328, Online. Association for Computational Linguistics.

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

Cyril Goutte, Serge Léger, and Marine Carpuat. 2014. The NRC System for Discriminating Similar Languages. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects (VarDial)*, pages 139–145, Dublin, Ireland.

Mihaela Găman, Dirk Hovy, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubešić, Niko Partanen, Christoph

Purschke, Yves Scherrer, and Marcos Zampieri. 2020. A Report on the VarDial Evaluation Campaign 2020. In *Proceedings of the Seventh Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–14.

Adeep Hande, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2020. KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 54–63, Barcelona, Spain (Online). Association for Computational Linguistics.

Arthur S House and Edward P Neuburg. 1977. Toward automatic identification of the language of an utterance. I. Preliminary methodological considerations. *The Journal of the Acoustical Society of America*, 62(3):708–713.

Hai Hu, Wen Li, He Zhou, Zuoyu Tian, Yiwen Zhang, and Liang Zou. 2019. Ensemble Methods to Distinguish Mainland and Taiwan Chinese. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 165–171.

Aaron Jaech, George Mulcaire, Mari Ostendorf, and Noah A Smith. 2016. A neural model for language identification in code-switched tweets. In *Proceedings of The Second Workshop on Computational Approaches to Code Switching*, pages 60–64.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2020a. Experiments in language variety geolocation and dialect identification. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 220–231.

Tommi Jauhiainen, Heidi Jauhiainen, Niko Partanen, and Krister Lindén. 2020b. Uralic Language Identification (ULI) 2020 shared task dataset and the Wanca 2017 corpora. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 173–185.

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 (VarDial 2016).*, pages 153–162.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Tim Kreutz and Walter Daelemans. 2018. Exploring classifier combinations for language variety identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 191–198.

Marco Lui and Timothy Baldwin. 2014. Accurate language identification of Twitter messages. In *Proceedings of the 5th workshop on language analysis for social media (LASM)*, pages 17–25.

Paul McNamee. 2005. Language identification: a solved problem suitable for undergraduate instruction. *Journal of computing sciences in colleges*, 20(3):94–101.

Maria Medvedeva, Martin Kroon, and Barbara Plank. 2017. When sparse traditional models outperform dense neural networks: the curious case of discriminating between similar languages. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 156–163, Valencia, Spain. Association for Computational Linguistics.

Cristian Popa and Vlad Stefănescu. 2020. Applying multilingual and monolingual transformer-based models for dialect identification. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 193–201.

Yves Scherrer and Nikola Ljubešić. 2020. HeLju@ VarDial 2020: Social media variety geolocation with BERT models. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 202–211.

Fernando Benites de Azevedo e Souza, Ralf Grubenmann, Pius von Däniken, Dirk Von Gruenigen, Jan Milan Deriu, and Mark Cieliebak. 2018. Twist bytes: German dialect identification with data mining optimization. In *27th International Conference on Computational Linguistics (COLING 2018), Santa Fe, August 20-26, 2018*, pages 218–227. VarDial.

Diana Tudoreanu. 2019. DTeam@ VarDial 2019: Ensemble based on skip-gram and triplet loss neural networks for Moldavian vs. Romanian cross-dialect topic identification. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 202–208.

Jason Wang and Luis Perez. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Nianheng Wu, Eric DeMattos, Kwok Him So, Pin-zhen Chen, and Çağrı Çöltekin. 2019. Language discrimination and transfer learning for similar languages: experiments with feature combinations and adaptation. In *Proceedings of the Sixth Workshop on NLP* *for Similar Languages, Varieties and Dialects*, pages 54–63.

George-Eduard Zaharia, Andrei-Marius Avram, Dumitru-Clementin Cercel, and Traian Rebedea. 2020. Exploring the power of Romanian BERT for dialect identification. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 232–241.

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)*, pages 1–15, Valencia, Spain.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shuon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–17, Santa Fe, USA.

Marcos Zampieri, Shervin Malmasi, Yves Scherrer, Tanja Samardžić, Francis Tyers, Miikka Silfverberg, Natalia Klyueva, Tung-Le Pan, Chu-Ren Huang, Radu Tudor Ionescu, Andrei Butnaru, and Tommi Jauhiainen. 2019. A Report on the Third VarDial Evaluation Campaign. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

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

A Appendix - Data augmentation

Following Wei and Zou (2019), we ran some data augmentation experiments on the dataset of the RDI task. We experimented with Random Swap (swap the position of two words in the sentence), Random Delection (remove one word in the sentence), and Random Insertion (insert one extra word in the sentence). We did not experiment with Random Replacement, which involves the replacement of a word with a synonym. We also experimented with an additional technique, Shuffling, in which the words of the sentence are simply shuffled, and which is essentially a variation of Random Swap. We trained the network, on each augmented dataset, for 5 different times, and determined its test accuracy on the same hold-out dataset.

The results of the experiments on the model pretrained on news (the news+tweets model) are in Table 14. All techniques led to an improvement of the performance of the network, and the best improvement was obtained by shuffling the words of the sentences.

Technique	Training Epochs	Macro F_1
Shuffling	**5**	**0.756**
Random Swap	5	0.733
Random Deletion	5	0.733
Random Insertion	5	0.727
No augmentation	10	0.709

Table 14: Summary of our experiments on data augmentation in the RDI task, on the news+tweets model.