Experiments in Language Variety Geolocation and Dialect Identification

Tommi Jauhiainen
Department of Digital Humanities
University of Helsinki
tommi.jauhiainen@helsinki.fi

Heidi Jauhiainen
Department of Digital Humanities
University of Helsinki
heidi.jauhiainen@helsinki.fi

Krister Lindén
Department of Digital Humanities
University of Helsinki
krister.linden@helsinki.fi

Abstract

In this paper we describe the systems we used when participating in the VarDial Evaluation Campaign organized as part of the 7th workshop on NLP for similar languages, varieties and dialects. The shared tasks we participated in were the second edition of the Romanian Dialect Identification (RDI) and the first edition of the Social Media Variety Geolocation (SMG). The submissions of our SUKI team used generative language models based on Naive Bayes and character *n*-grams.

1 Introduction

We first took part in the related language identification shared tasks in 2015 (Jauhiainen et al., 2015) and we have been using the same team name *SUKI* ever since. The shared tasks have been organized as part of the VarDial workshops dealing with computational methods and language resources for closely related languages, language varieties, and dialects. The 2020 VarDial Evaluation Campaign contained three separate shared tasks (Găman et al., 2020).[1] We participated in the Romanian Dialect Identification (RDI) and the Social Media Variety Geolocation (SMG) shared tasks. We did not participate in the third task, Uralic Language Identification (ULI), as we were part of the team organizing it (Jauhiainen et al., 2020).

In this paper, we first introduce some previous work related to these shared tasks, to language identification and to identification of Romanian dialects in particular as well as to geolocation of texts. Then we describe the RDI and the SMG shared tasks, their datasets and the systems we used in our submissions as well as the results of the shared tasks.

2 Related work

2.1 Shared tasks

The RDI and the SMG shared tasks were organized as a part of the VarDial Evaluation Campaign 2020, which continued the tradition of shared tasks focusing on close languages for the seventh consecutive year (Zampieri et al., 2014; Zampieri et al., 2015; Malmasi et al., 2016; Zampieri et al., 2017; Zampieri et al., 2018; Zampieri et al., 2019). The RDI shared task was a continuation of the first track of the Moldavian vs. Romanian Cross-dialect Topic identification (MRC) shared task organized in 2019 (Zampieri et al., 2019). The SGM was the first shared task of its kind and the first language identification shared task where the aim was to pin a text to a location.

[1] https://sites.google.com/view/vardial2020/evaluation-campaign

Proceedings of the 7th VarDial Workshop on NLP for Similar Languages, Varieties and Dialects, pages 220–231
Barcelona, Spain (Online), December 13, 2020

2.2 Language identification in texts

Automatic language identification in texts was first introduced in the 1960s (Mustonen, 1965). A recent survey by Jauhiainen et al. (2019d) introduces the different aspects of language identification as well as most of the methods used for it during the past 50 years. The methods used for the task of language identification are mostly shared with other classification tasks as almost any modern machine learning method can be trained to distinguish between different languages (Jauhiainen, 2019). Support Vector Machines (SVM) are among the most popular and successful machine learning algorithms that have been applied to language identification and have traditionally been very competetive in language identification shared tasks, winning many of them (Goutte et al., 2014; Malmasi and Dras, 2015; Çöltekin and Rama, 2016; Malmasi and Zampieri, 2016; Bestgen, 2017; Malmasi and Zampieri, 2017; Çöltekin et al., 2018; Wu et al., 2019). Deep learning methods have traditionally been less successful in language identification than in other classification tasks (Çöltekin and Rama, 2016; Gamallo et al., 2016; Medvedeva et al., 2017). The first time a language identification shared task was won using deep learning was in the Cuneiform Language Identification (CLI) shared task we organized in 2019 (Zampieri et al., 2019; Jauhiainen et al., 2019a) when it was won by Bernier-Colborne et al. (2019) using BERT-based classifier (Devlin et al., 2018).

In our submissions for the shared tasks, we used language identifiers based on the product of relative frequencies we had developed for the VarDial Evaluation campaign of the previous year (Jauhiainen et al., 2019a; Jauhiainen et al., 2019b). The method is basically the same as Naive Bayes (NB) using the observed relative frequencies of character n-grams as probabilities.

2.3 Identification of Romanian dialects

The RDI shared task focused on distinguishing between Moldavian and Romanian. Romanian and Moldavian are coupled together as dialects of the Romanian language (ron) in the ISO 639-3 standard (SIL, 2020).[2]

The shared task debuted as one of the tracks of the MRC shared task of the VarDial Evaluation Campaign 2019 (Zampieri et al., 2019). The aim was to maximize the macro-averaged F_1 score for the two dialects. When macro-averaging, the F_1 score of each individual dialect is calculated first and the result is the average of those F_1 scores. The dataset of the shared task was published as the Moldavian and Romanian Dialectal Corpus (MOROCO) (Butnaru and Ionescu, 2019).

The 2019 edition was officially won by Tudoreanu (2019) using two character-level neural networks which were combined as an ensemble using SVM in the manner of Stacked Generalisation (Wolpert, 1992). Some of the participants had problems producing the correct number of lines for their submissions (Zampieri et al., 2019) and produced corrected results after the end of the shared task for their system description papers (Onose et al., 2019; Wu et al., 2019).

Some additional experiments using the MOROCO data have also been reported (Tudoreanu, 2019; Onose et al., 2019; Găman and Ionescu, 2020; Georgescu et al., 2020). Găman and Ionescu (2020) conducted an evaluation of the data and compared the performance of several methods with the annotations done by native speakers of the dialects. They noted that the machine learning methods were superior to humans in distinguishing between the two dialects and concluded that the models were better at finding character level clues than human annotators.

The results of all these experiments were not available together, so we collected them in Table 1. The first column describes the method used as well as the possible MRC team name in parentheses. The second column gives the Macro F_1 score and the third lists the source for the information. The methods used include SVMs, Kernel Ridge Regression (KRR), Convolutional Neural Networks (CNN), Hierarchical Attention Networks (HAN) (Yang et al., 2016), Bidirectional Gated Recurrent Units (BiGRU), Long Short-Term Memory cells (LSTM), and Recurrent Neural Networks (RNN). The MRC shared task was supposed to be closed, i.e. no task external data was to be used. However, pre-trained word vectors were used during the competition and have also been used in some of the later experiments. The pre-trained word vectors used are from Common Crawl (Grave et al., 2018), Romanian Language Corpus (CoRoLa)

[2] https://iso639-3.sil.org/code/ron

(Mititelu et al., 2018), and the Nordic Language Processing Laboratory "NLPL" (Kutuzov et al., 2017) and they have been indicated in Table 1.

Method	**Macro F_1**	**Reported by**
Linear SVM classifier with LM adaptation (tearsofjoy)	0.962	Wu et al. (2019)
Stacking with 12 classifiers	0.945	Găman and Ionescu (2020)
KRR with string kernels	0.943	Găman and Ionescu (2020)
KRR	0.941	Butnaru and Ionescu (2019)
CNN+SE with ADA activation	0.940	Georgescu et al. (2020)
SVM with string kernels	0.939	Găman and Ionescu (2020)
CNN with ADA activation	0.937	Georgescu et al. (2020)
CNN+SE+PyNADA with ReLU and ADA activations	0.937	Georgescu et al. (2020)
CNN with ReLU and ADA activations	0.936	Georgescu et al. (2020)
Ensemble (DTeam)	0.934	Tudoreanu (2019)
Neural network based on softmax loss (DTeam)	0.933	Tudoreanu (2019)
CNN+SE with leaky ReLU activation	0.931	Georgescu et al. (2020)
HAN with FastText Common Crawl word vectors (SC-UPB)	0.930	Onose et al. (2019)
CNN+SE with ReLU activation	0.930	Georgescu et al. (2020)
CNN+SE	0.929	Butnaru and Ionescu (2019)
CNN with characters	0.929	Găman and Ionescu (2020)
Voting between 12 classifiers	0.929	Găman and Ionescu (2020)
CNN with leaky ReLU activation	0.929	Georgescu et al. (2020)
CNN (DTeam)	0.928	Tudoreanu (2019)
CNN with ReLU activation	0.928	Georgescu et al. (2020)
CNN	0.927	Butnaru and Ionescu (2019)
BiGRU with FastText Common Crawl word vectors (SC-UPB)	0.903	Onose et al. (2019)
Two skip-gram CNNs stacked using SVM (DTeam)	*0.895*	*Zampieri et al. (2019)*
LSTM with CoRoLa word vectors	0.895	Găman and Ionescu (2020)
Neural network based on triplet loss (DTeam)	0.869	Tudoreanu (2019)
BiGRU with CoRoLa word vectors	0.868	Onose et al. (2019)
BiGRU with Common Crawl word vectors	0.865	Găman and Ionescu (2020)
LSTM with NLPL word vectors	0.852	Găman and Ionescu (2020)
LSTM with FastText Common Crawl word vectors (SC-UPB)	0.847	Onose et al. (2019)
BiGRU with NLPL word vectors (SC-UPB)	0.834	Onose et al. (2019)
LSTM with CoRoLa word vectors (SC-UPB)	0.825	Onose et al. (2019)
LSTM with NLPL word vectors (SC-UPB)	0.798	Onose et al. (2019)
Majority voting between 5 classifiers on 40 features (R2I_LIS) (train+dev)	*0.796*	*Zampieri et al. (2019)*
Majority voting between 5 classifiers on 40 features (R2I_LIS) (train+dev)	0.778	Chifu (2019)
Majority voting between 5 classifiers on 40 features (R2I_LIS) (train)	0.776	Chifu (2019)
Linear SVM classifier (tearsofjoy)	*0.757*	*Zampieri et al. (2019)*
Word-level bigrams with add-one smoothing (lonewolf)	*0.735*	*Zampieri et al. (2019)*
RNN with GRUs and pre-trained FastText model (SC-UPB)	*0.709*	*Zampieri et al. (2019)*
HAN with CoRoLa word vectors	0.697	Găman and Ionescu (2020)
HAN with NLPL word vectors	0.694	Găman and Ionescu (2020)
HAN with Common Crawl word vectors	0.694	Găman and Ionescu (2020)
Character-level bigrams with add-one smoothing (lonewolf)	0.656	Chifu (2019)
Word-level bigrams with Good-Turing smoothing (lonewolf)	0.608	Chifu (2019)
HAN with FastText Common Crawl word vectors (SC-UPB)	0.508	Onose et al. (2019)

Table 1: The Macro F_1 scores for Romanian dialect identification on the MOROCO dataset reported by various papers. The official submissions of each team is in italics.

2.4 Geolocation of texts

Many datasets of social media texts come with some sort of location attached to each text. People living close to each other tend to speak and write in similar ways and about common subjects. This makes identification of the location of a tweet[3] or a jodel[4] possible by just looking at the text produced. In addition to a text itself, many geolocation detection methods use, for example, the metadata of the user profile (Huang and Carley, 2017) or other texts produced by the same user (Chong and Lim, 2019). In the context of the SMG shared task, the aim was to distinguish dialectal differences based on just the text of the tweet or jodel itself. In many studies, the aim has been to pinpoint a correct city (Huang and Carley, 2017; Snyder et al., 2019), a country (Huang and Carley, 2017), or even a specific venue, like

[3]https://twitter.com

[4]https://jodel.com

a restaurant or a shop (Chong and Lim, 2019). In the SMG shared task the exact coordinates for each mystery text were required.

3 Romanian Dialect Identification (RDI) shared task

The RDI shared task was a two-way classification task between texts written in the Moldavian or Romanian dialects of the Romanian language. Each participating team was allowed to submit three runs and the runs were evaluated based on the macro-averaged F_1 score. We experimented with three classification techniques and in the end we submitted only one run to the shared task gaining fourth place among the eight teams submitting results.

3.1 Test setup

In order to train their models, the participants were provided with the MOROCO data set (Butnaru and Ionescu, 2019). The original MOROCO data set is divided into training, validation, and testing, but for this shared task, all the 33,564 samples of text were to be used for training. All in all, there were 15,403 texts for Moldavian and 18,161 texts for Romanian.

For validation, two different sets of texts were provided. The first validation set *dev-source* was in-domain with the training data (texts were from the news domain) while the second validation set *dev-target* was out-of domain (the texts were tweets). The *dev-source* contained 2,718 additional texts for Moldavian and 3,205 texts for Romanian. However, the *dev-target* was considerably smaller with only 113 texts for Moldavian and 102 texts for Romanian.

The test set included 5,022 lines of texts without language labels. The participants were informed that they were tweets similar to those of the second validation set *dev-target*. The average length of a line was 98 characters. In all of the three datasets, the named entities had been transformed to "NE" tags. Butnaru and Ionescu (2019) do not specify how the named entity removal was performed in practice or whether it was automatic or manual.

3.2 Experiments on the development set

The participants were informed beforehand that the shared task test set would consist of tweets. For this reason, we focused our experiments on the out-of-domain validation set *dev-target*. Thus, we combined the MOROCO dataset and the in-domain validation data *dev-source* as the training data for our experiments.

For this task, we set out to experiment with the same methods as we did for the baseline of the CLI shared task in 2019 (Jauhiainen et al., 2019a). We used the product of relative frequencies method and its adaptive version also in the German Dialect Identification (GDI) and Discriminating between the Mainland and Taiwan variation of Mandarin Chinese (DMT) shared tasks of the VarDial Evaluation campaign 2019 (Jauhiainen et al., 2019b). With the adaptive version, we won the DMT track for traditional Chinese (Zampieri et al., 2019). Without adaptation, the results of our implementation of the NB classifier were comparable to other methods not using language model adaptation, e.g. SVM ensembles or deep neural networks such as RNN, CNN, and LSTM (Jauhiainen et al., 2019b). Unfortunately, in this years campaign, we did not have time to experiment with the HeLI method (Jauhiainen et al., 2016).

For the CLI task, one of the methods produced best results using character *n*-grams up to 15 characters and we started experimenting using similar very long character sequences. It was soon evident, that our server environment was not capable of processing the training data with such long *n*-grams and we ended up using a maximum of 12 character sequences.[5] The three methods we evaluated on the RDI training and development data were the sum of relative frequencies, the simple-scoring, and the product of relative frequencies methods. In all of the three methods, the language models for the two dialects consist of all the possible character *n*-grams extracted from their training data.

3.2.1 Sum of relative frequencies

In the sum of relative frequencies, the character *n*-grams extracted from the mystery text to be identified M are compared with the language models $dom(O(C_g))$ of the two dialects g, and for each *n*-gram found

[5]The problem was mainly with the memory usage: we used a maximum of 60 gigabytes as our Java memory heap.

in a language model, the score $R_{sum}(g, M)$ is increased by the respective relative frequency. The dialect gaining the highest score is selected. Jauhiainen et al. (2019d) formulate the method as in Equation 1:

$$R_{sum}(g, M) = \sum_{i=1}^{l_{MF}} \frac{c(C_g, f_i)}{l_{C_g^F}} \quad (1)$$

where l_{MF} is the number of individual features in the text M and $c(C_g, f_i)$ is the count of its ith feature f_i in the training corpus.

All the possible combinations of character *n*-grams from 1 to 12 were evaluated. The best results with a macro F_1 of 0.4848 were obtained using only character 12-grams. The score was very low considering that the task was a simple binary classification between two dialects.

3.2.2 Simple scoring

In simple scoring, the character *n*-grams from the text to be identified (M) are compared with the language models and the scores of the dialects are increased by one for each one found. The dialect with the highest score is predicted. Jauhiainen et al. (2019d) formulate the method as in Equation 2:

$$R_{simple}(g, M) = \sum_{i=1}^{l_{MF}} \begin{cases} 1 & \textbf{, if } f_i \in dom(O(C_g)) \\ 0 & \textbf{, otherwise} \end{cases} \quad (2)$$

Again we experimented with all the possible combinations of character *n*-grams from 1 to 12. The results were much more promising than when using the sum of relative frequencies method. Table 2 shows some of the best character *n*-gram combinations from these experiments. The column "*n* Min" tells the minimum length of the character *n*-grams used and the "*n* Max" the maximum length.

***n* Min**	***n* Max**	**Macro F_1**
1-2	12	0.6099
3 or 5	12	0.6142
4	12	0.6193
1-3	11	0.6193
4	11	0.6099
1-3	10	0.5978

Table 2: Experiments with the simple scoring method on the RDI validation target.

As can be noticed, four combinations gave exactly the same highest result of 0.6193. The reason for arriving at the exact same score was that the *dev-target* was a relatively small set only totaling 215 texts.

3.2.3 Product of relative frequencies

The third method evaluated was the product of relative frequencies. We used the same implementation of this method in last years Evaluation Campaign (Jauhiainen et al., 2019a; Jauhiainen et al., 2019b). In this method, the relative frequencies are multiplied together. Jauhiainen et al. (2019d) formulate the method as in Equation 3:

$$R_{prod}(g, M) = \prod_{i}^{l_{MF}} \frac{c(C_g, f_i)}{l_{C_g^F}} \quad (3)$$

In case the feature f_i was not found in the training corpus of a dialect C_g, a smoothing value was used. The smoothing value was the relative frequency of a feature found only once in the training corpus. In the actual implementation of the algorithm, the sum of negative logarithms of the relative frequencies were used. The smoothing value was multiplied by a penalty modifier determined using the development set.

We experimented with several combinations of values for the minimum and maximum lengths of character *n*-grams as well as the penalty modifiers. The best results were obtained using character *n*-grams from five to eigth with a penalty modifier of 1.18. These results together with the results of some nearby parameter combinations can be seen in Table 3. The results using the product of relative

frequencies were clearly superior to the results of the two previous methods so we decided to do some more experiments with this method.

***n* Min**	***n* Max**	**Penalty modifier**	**Macro F_1**
5	8	1.18	0.6528
5	8	1.19	0.6485
5	8	1.17	0.6475
4	8	1.25-1.26	0.6451
5	9	1.25-1.29	0.6451

Table 3: Experiments with the product of relative frequencies method on the RDI validation target.

So far the results were obtained using the training and validation sets without any preprocessing. First we experimented with removing the "NE" tags representing named entities from the datasets before training and evaluation. There was a clear increase in the F_1 score as evidenced in Table 4.

***n* Min**	***n* Max**	**Penalty modifier**	**Macro F_1**
4	7	1.14	0.6706
4	7	1.11-1.13	0.6656
3	7	1.15-1.18	0.6656
4	7	1.15	0.6663
4	8	1.12	0.6598

Table 4: Experiments with the product of relative frequencies method on the RDI validation target with the named entity tags removed.

We further experimented with removing all non-alphabetic characters, which again increased the F_1-score (Table 5). As "alphabetic" characters we considered all characters included in the character set of any language according to Java regular expressions.[6]

***n* Min**	***n* Max**	**Penalty modifier**	**Macro F_1**
4	5	1.35-1.36	0.6877
4	6	1.40-1.48	0.6836
4	5	1.37-1.39	0.6832
4	5	1.34	0.6829
3	5	1.40-1.42	0.6648

Table 5: Experiments with the product of relative frequencies method on the RDI validation target using only alphabetic characters.

The next preprocessing step was lowercasing all characters (Table 6). This ended up giving a further boost of 2.1% to the Macro F_1 score. All in all, using these three simple pre-processing steps increased the F_1 score by 7.6%.

In the two previous VarDial Evaluation campaigns we were able to gain good results using language model adaptation together with the product of relative frequencies and the HeLI methods (Jauhiainen et al., 2018a; Jauhiainen et al., 2018b; Jauhiainen et al., 2019b; Jauhiainen et al., 2019c). We did evaluate using the product of relative frequencies with the best parameters from the previous trials together with language model adaptation. This combination did not further increase the score, instead with various parameters the scores were actually lower. Thus, we decided not to use language model adaptation in the actual run.

3.3 Results

We removed the named entity tags from the test set and used only lowercased alphabetic characters. We ended up with two lines for which our language identifier returned an unknown language. This was due to both lines being a single "a" after preprocessing and the classifier using a minimum length of four for the character *n*-grams. We changed them to Moldavian (MD) as we were not allowed to submit any

[6]For the exact regular expression see line 241 of HeLI.java at https://github.com/tosaja/HeLI/blob/master/HeLI.java

n Min	*n* Max	Penalty modifier	Macro F_1
4	5	1.39-1.41	0.7023
4	5	1.33-1.38 and 1.42-1.48	0.6976
3	5	1.21-1.22	0.6976
4	6	1.21-1.24	0.6879

Table 6: Experiments with the product of relative frequencies method on the RDI validation target using only lowercased alphabetic characters.

other tags than one of the two Romanian dialects. We submitted these results as our only run to the RDI shared task.

Rank	Team	run	Macro F_1
1	Tubingen	1	0.7876
2	Anumiti	3	0.7751
3	Phlyers	1	0.6661
4	**SUKI**	**1**	**0.6584**
5	UPB	1	0.6476
6	UAIC	1	0.5550
7	akanksha	1	0.4813
8	The_Linguistadors	2	0.4294

Table 7: The best results of each team participating on the RDI 2020 shared task.

Our final score of 0.6584 on the test set was in line with what can be expected from gaining an F-score around 0.70 on the development set. The score gave us the fourth place in the shared task (Table 7).

4 Social Media Variety Geolocation (SMG) shared task

This task was divided into three separate tracks, each focusing on its own geographic area. The first track, DE-AT, included Jodel conversations in standard German from Germany and Austria. Second track, CH, focused on Swiss German Jodel conversations from Switzerland. The third track, BCMS, featured tweets from Croatia, Bosnia and Herzegovia, Montenegro, and Serbia. All the tracks were scored using the median distance in kilometers of all the predicted locations to the actual ones.

4.1 The datasets for the shared task

Before the testing period the participants were provided with training and development data for each of the tracks. The data for the DE-AT and CH tracks came from a mobile chat application called Jodel, where users can chat anonymously with others in the same area (Hovy and Purschke, 2018) while the data for the BCMS track consisted of tweets (Ljubešić et al., 2016). The sizes of each dataset are shown in Table 8. The reason why the BCMS data is divided between many more locations than the DE-AT and CH data is that the coordinates of the locations are given in much more detail (with up to 8 decimal places in BCMS and only 2 decimal places for DE-AT and CH).

Track	Set	#Texts	Average length in tokens	Number of unique locations
DE-AT	Training	336,983	71	5,228
DE-AT	Development	46,582	71	6,512
DE-AT	Test	48,239	69	???
CH	Training	22,600	55	222
CH	Development	3,068	57	339
CH	Test	3,097	55	???
BCMS	Training	320,042	13	264,741
BCMS	Development	39,750	13	36,992
BCMS	Test	39,723	13	???

Table 8: The sizes of the datasets for the SMG shared task.

4.2 The methods used

We opted for a simple approach of dividing the given geographic areas into 81 equally sized geographical areas. For both longitude and latitude, the distance between maximum and minimum points in the datasets was divided by 9. Each of the 100 unique corner coordinates of the 81 areas functioned as a gathering point. All the jodels or tweets in the training data were gathered at their nearest gathering point. After that the location of each gathering point was adjusted to be in the center of the original positions of the tweets or jodels gathered at the point. We only had resources to try different divisions with the Swiss German data and we experimented with dividing by 8 or 10 instead of 9, but this did not improve the results.

Not all of the points gathered texts, but for all those that included texts, a language model was created. That language model was then used with a language identifier. While experimenting, we calculated our results using average distance instead of median distance. The distance we used was based on coordinate points and as the longitude and latitude are not equal in kilometers, our optimization was not perfect. We used the same product of relative frequencies classifier in language identification, which was described earlier with regard to the RDI shared task (Equation 3). The parameters we used with each track can be seen in Table 9.

Track	*n* Min	*n* Max	Penalty modifier
DEAT	1	5	1.9
CH	2	3	2.35
BCMS	1	7	3.5

Table 9: Parameters used in the submissions for the three SMG tracks.

4.3 Results

Table 10 shows the results of the teams participating in the SMG 2020 shared task DEAT track. Our submission was clearly the least efficient of all submitted systems.

Rank	Team	Median distance	Mean distance
1	helsinki-ljubljana	159.59	183.97
2	Piyush_Mishra	183.99	204.93
3	CUBoulder-UBC	198.27	218.51
4	ZHAW	205.81	230.78
5	**SUKI**	**243.12**	**266.85**

Table 10: The best results of each team participating in the SMG 2020 shared task DEAT track.

Table 11 shows the results of the teams participating in the SMG 2020 shared task Swiss German track.

Rank	Team	Median distance	Mean distance
1	ZHAW	15.93	25.06
2	helsinki-ljubljana	17.66	26.21
3	CUBoulder-UBC	19.49	27.63
4	**SUKI**	**23.96**	**34.59**
5	UnibucKernel	25.57	30.52
6	The_lingustadors	26.70	31.21
7	Piyush_Mishra	27.31	33.20

Table 11: The best results of each team participating in the SMG 2020 shared task Swiss German track.

Table 12 shows the results of the teams participating in the SMG 2020 shared task BCMS track. Our submission gave us the third position which was our best ranking among the tracks. The winning helsinki-ljubljana teams median distance was in a league of its own, but we were relatively close to the ZHAW teams result.

Rank	Team	Median distance	Mean distance
1	helsinki-ljubljana	48.99	86.83
2	ZHAW	57.24	100.42
3	**SUKI**	**61.01**	**105.11**
4	CUBoulder-UBC	64.76	106.67
5	Piyush_Mishra	85.70	112.65
6	The_lingustadors	97.16	141.88

Table 12: The best results of each team participating on the SMG 2020 shared task BCMS track.

5 Conclusions and future work

In this paper, we presented the systems we experimented with when participating in two of the shared tasks organized as part of the VarDial Evaluation Campaign 2020. Our systems did not reach the state of the art in any of the tracks of the shared tasks.

There is clearly some room for improvement in the approaches we used. We did not have time to experiment with using adaptive language models in the SMG shared task. Using them might have improved the identification accuracy considerably. Also, using 100 coordinate points might not have been optimal in the DEAT and BCMS tracks as the areas where those texts came from were larger than the CH track but we only optimized the division using the CH track training data. There is also room for improvement in how we optimized the parameters for the SMG shared task as we used the average distance and the distance "unit" we used were coordinate points instead of kilometers.

References

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.

Yves Bestgen. 2017. Improving the Character Ngram 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)*, pages 115–123, Valencia, Spain.

Andrei Butnaru and Radu Tudor Ionescu. 2019. Moroco: The Moldavian and Romanian dialectal corpus. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 688–698.

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

Adrian-Gabriel Chifu. 2019. The R2I_LIS team proposes majority vote for VarDial's MRC task. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 138–143.

Wen-Haw Chong and Ee-Peng Lim. 2019. Fine-grained geolocation of tweets in temporal proximity. *ACM Transactions on Information Systems (TOIS)*, 37(2):1–33.

Çağrı Çöltekin, Taraka Rama, and Verena Blaschke. 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)*, pages 55–65.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Mihaela Găman and Radu Tudor Ionescu. 2020. The unreasonable effectiveness of machine learning in Moldavian versus Romanian dialect identification. *arXiv preprint arXiv:2007.15700*.

Mariana-Iuliana Georgescu, Radu Tudor Ionescu, Nicolae-Catalin Ristea, and Nicu Sebe. 2020. Non-linear neurons with human-like apical dendrite activations. *arXiv preprint arXiv:2003.03229*.

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.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Language Resources and Evaluation Conference*, number CONF.

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)*.

Dirk Hovy and Christoph Purschke. 2018. Capturing regional variation with distributed place representations and geographic retrofitting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4383–4394, Brussels, Belgium. Association for Computational Linguistics.

Binxuan Huang and Kathleen M Carley. 2017. On predicting geolocation of tweets using convolutional neural networks. In *International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation*, pages 281–291. Springer.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2015. Discriminating Similar Languages with Token-based Backoff. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 44–51, Hissar, Bulgaria.

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

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2018a. HeLI-based Experiments in Swiss German Dialect Identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 254–262, Santa Fe, NM.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 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)*, pages 66–75, Santa Fe, NM.

Tommi Jauhiainen, Heidi Jauhiainen, Tero Alstola, and Krister Lindén. 2019a. Language and dialect identification of cuneiform texts. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 89–98. Association for Computational Linguistics.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2019b. Discriminating between Mandarin Chinese and Swiss-German varieties using adaptive language models. In *Proceedings of the 6th Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2019)*, pages 178–187, Minneapolis, Minnesota.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2019c. Language model adaptation for language and dialect identification of text. *Natural Language Engineering*, 25(5):561–583.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019d. Automatic Language Identification in Texts: A Survey. *Journal of Artificial Intelligence Research*, 65:675–782.

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

Tommi Jauhiainen. 2019. *Language identification in texts*. Ph.D. thesis, University of Helsinki, Finland.

Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 58th Conference on Simulation and Modelling*, pages 271–276. Linköping University Electronic Press.

Nikola Ljubešić, Tanja Samardžić, and Curdin Derungs. 2016. TweetGeo - a tool for collecting, processing and analysing geo-encoded linguistic data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3412–3421, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Shervin Malmasi and Mark Dras. 2015. Language Identification using Classifier Ensembles. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 35–43, Hissar, Bulgaria.

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

Shervin Malmasi and Marcos Zampieri. 2017. German Dialect Identification in Interview Transcriptions. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 164–169, Valencia, Spain.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating Between Similar Languages and Arabic Dialect Identification: A Report on the Third DSL Shared Task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–14, Osaka, Japan.

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.

Verginica Barbu Mititelu, Dan Tufiş, and Elena Irimia. 2018. The reference corpus of the contemporary Romanian language (CoRoLa). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Seppo Mustonen. 1965. Multiple Discriminant Analysis in Linguistic Problems. *Statistical Methods in Linguistics*, 4:37–44.

Cristian Onose, Dumitru-Clementin Cercel, and Stefan Trausan-Matu. 2019. SC-UPB at the VarDial 2019 evaluation campaign: Moldavian vs. Romanian cross-dialect topic identification. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 172–177.

SIL. 2020. *ISO 639-3 Codes for the representation of names of languages*. SIL International.

Luke S Snyder, Morteza Karimzadeh, Ray Chen, and David S Ebert. 2019. City-level geolocation of tweets for real-time visual analytics. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pages 85–88.

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.

David H. Wolpert. 1992. Stacked Generalization. *Neural Networks*, 5(2):241–259.

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.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A Report on the DSL Shared Task 2014. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 58–67, Dublin, Ireland.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the DSL Shared Task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 1–9, Hissar, Bulgaria.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešic, 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*, pages 1–15, Valencia, Spain.

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

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.