Naive Bayes-based Experiments in Romanian Dialect Identification

Tommi Jauhiainen[1,2], **Heidi Jauhiainen**[1], **Krister Lindén**[1]
[1]Department of Digital Humanities, University of Helsinki, Finland
[2]Department of English, Rochester Institute of Technology, USA
tommi.jauhiainen@helsinki.fi

Abstract

This article describes the experiments and systems developed by the SUKI team for the second edition of the Romanian Dialect Identification (RDI) shared task which was organized as part of the 2021 VarDial Evaluation Campaign. We submitted two runs to the shared task and our second submission was the overall best submission by a noticeable margin. Our best submission used a character *n*-gram based naive Bayes classifier with adaptive language models. We describe our experiments on the development set leading to both submissions.

1 Introduction

Romanian Dialect Identification (RDI) 2021 was a shared task focusing on discriminating between two written language varieties, Romanian and Moldavian. These are considered dialects of the Romanian language (ron) by the ISO-639-3 standard.[1] Together with the Uralic Language Identification (ULI) shared task it was one of the newest additions to a series of shared tasks in language or dialect identification organized as parts of the VarDial workshop series (Zampieri et al., 2014, 2015; Malmasi et al., 2016; Zampieri et al., 2017, 2018, 2019; Gaman et al., 2020; Chakravarthi et al., 2021).

Since the 2015 edition of VarDial, we have participated in many of the language identification shared tasks as the SUKI team. In the beginning, we were developing the original HeLI method (Jauhiainen, 2010; Jauhiainen et al., 2016), then we added language model adaptation to it (Jauhiainen et al., 2018a), and in this article we aim to improve on the results of our implementation of a naive Bayes based classifier (Jauhiainen et al., 2020) using character *n*-grams with and without language model adaptation (Jauhiainen et al., 2019a).

In this article, we first give a quick overview of the related work on language identification, romanian dialect identification in particular, blacklist classifiers, and language model adaptation in Section 2. In Section 3, we provide information about the data set and the evaluation measures used in the RDI shared task. Then, in Section 4, we proceed to describe our experiments in building a competitive Romanian dialect identification system using the provided training data. In the experiment section, we also describe the methods used in sufficient detail for the reader to reproduce the systems. Sections 5 and 6 provide the results we attained in the shared task and the conclusions we were able to make from them and from our experiments on the training data.

2 Related Work

Dialect identification is part of a broader task of language identification. Automatic language identification of digital texts has been investigated for over 50 years. The first published experiments where a computer program has identified languages were conducted by Mustonen (1965). Identification of a long, monolingual, text is very simple if the languages to be identified are clearly different from one another. However, the task of language identification includes many challenges which increase its difficulty. For example, the texts can be very short, the languages to be discriminated can be very similar, or the amount of training data can be very small. When these challanges occur together, they can make language identification extremely difficult. The methods used and the challenges faced so far in language identification research were compiled in an extensive survey by Jauhiainen et al. (2019b).

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

Proceedings of the 8th VarDial Workshop on NLP for Similar Languages, Varieties and Dialects, pages 76–83
April 20, 2021 ©2021 Association for Computational Linguistics

2.1 Romanian Dialect Identification

RDI 2021 is the third shared task focusing on discriminating between the Moldavian and Romanian dialects of the Romanian language (ron). It appeared for the first time as one of the tracks of the Moldavian vs. Romanian Cross-dialect Topic identification shared task (MRC) in 2019 (Zampieri et al., 2019) and as a separate task in 2020 (Gaman et al., 2020).

We participated in the RDI 2020 shared task with a naive Bayes based dialect identifier. In our system description article for RDI 2020 (Jauhiainen et al., 2020), we provide an extensive listing of previous dialect identification experiments using the MOROCO dataset (Butnaru and Ionescu, 2019), which also functions as the training set for the shared task at hand. The highest F_1 score so far presented for the MOROCO dataset, 0.962, was provided by Wu et al. (2019) using a linear SVM classifier with language model adaptation.

Next we will shortly introduce some of the systems used in the RDI 2020 edition (Gaman et al., 2020). The RDI 2020 shared task was won by Çöltekin (2020). Her winning system, which obtained a Macro F_1 score of 0.7876, was based on a linear SVM classifier and used a language model adaptation scheme identical to the one used by Wu et al. (2019) a year earlier. In the winning system, she used only the provided target-development data for training the SVM. The best submission of the second ranked team, Anumiti (Popa and Ștefănescu, 2020), achieved an F_1 score of 0.7752. Their best system was an ensemble of five transformer-based classifiers combined using an SVM. The classifiers used in the ensemble were multilingual BERT, XLM, XLM-R, cased Romanian BERT, and uncased Romanian BERT. Ceolin and Zhang (2020) achieved a score of 0.6661 with a multinomial naive Bayes classifier using character n-grams from five to eight. We used a naive Bayes classifier with character n-grams from four to five, obtaining a Macro F_1 score of 0.6584 (Jauhiainen et al., 2020). Zaharia et al. (2020) used pre-trained Romanian BERT models and achieved an F_1 score of 0.6576. The UAIC team used an ensemble of TF-IDF encoders and a convolutional neural network attaining a score of 0.5553 (Rebeja and Cristea, 2020).

2.2 Blacklist Classifiers

Blacklists are lists of words or any other character sequences which if found, rule out the language. Blacklist classifiers have been previously used in language identification by Ljubešić et al. (2007) and Tiedemann and Ljubešić (2012). They used a blacklist of words with frequency cut-offs to distinguish between Bosnian, Croatian, and Serbian. Later, Barbaresi (2016) and Kroon et al. (2018) experimented with blacklisting when preparing for shared task submissions, but their experiments did not improve results and were not used in the final systems.

2.3 Language Model Adaptation

In language model adaptation, the language models are dynamically modified during the identification phase using the information in the mystery text (or collection of texts) itself. Using adaptive models in language identification shared tasks has proved to be important since we introduced the first language model adaptation experiments in language identification of texts during the VarDial Evaluation Campaign 2018 (Jauhiainen et al., 2018a,b; Zampieri et al., 2018).

As previously mentioned the winning submission of RDI 2020 used language model adaptation with SVMs (Çöltekin, 2020). The method used was identical to the one used by Wu et al. (2019) a year earlier in the shared tasks of the VarDial Evaluation Campaign 2019 (Zampieri et al., 2019). Using these systems, Wu et al. (2019) won the German Dialect Identification (GDI) 2019 shared task and the simplified track of the Discriminating between Mainland and Taiwan variation of Mandarin Chinese (DMT). In the method, the SVM was adapted once after the initial identification of all the test lines. The lines with a distance of 0.50 or higher from the SVM decision boundary were added to the training data of the identified language. The SVM was then re-trained with the additional data and re-run on the test set.

3 Shared Task Setup

The task of the RDI 2021 was to identify the dialect, either Romanian or Moldavian from a line of text.

The participants were provided with two sets of data, one for training and one for development. An augmented MOROCO data set (Butnaru and Ionescu, 2019) was used as the training data. The provided *train.txt* file included a total of 39,487 samples of texts from the news domain, which is 5,923 samples more than the MOROCO data set currently available on GitHub.[2]

The development data provided were tweets which in the RDI 2020 were used as the target development data (215 tweets) and the test data (5,022 tweets). The data set was published by Găman and Ionescu (2020) as MOROCO-Tweets and are available on GitHub.[3]

The participants were informed that the test set would also consist of tweets making the development set in-domain and the training set out-of-domain in relation to the test set. The task was closed, so only the data provided by the organizers was to be used in preparation of the participating systems. One of the motivations of the task was to evaluate cross-domain language identification systems but, nevertheless, the use of the development data for training the classifiers was not prohibited. The task description was very similar to the one in 2020. However, last year there were only 215 tweets available for development as opposed to the 5,237 tweets available for the 2021 edition.

In both the training and the development data, the named entities had been replaced with *NE* tags. The participants were informed that this would also be the case with the forthcoming test data. The development data included 2,612 lines of Moldavian and 2,625 lines of Romanian.

The test set was provided in the beginning of the general evaluation period of the VarDial Evaluation Campaign. The exact division of the samples between the two dialects in the test set was not a priori known to the participants. The test set contained 2,665 samples of Moldavian and 2,617 samples of Romanian which means that there was slightly larger difference between the two dialects than in the development set. As in RDI 2020, the evaluation measure used was the macro F_1 score. It gives both dialects equal value despite the possibility of there being an unequal number of samples in each dialect.

4 Experiments on the Development Data

When planning to participate in the RDI shared task, we contemplated evaluating two new additional methods combined with our existing language identifiers. The first additional method considered was active language model adaptation based on the length of the text to be identified (mystery text). However, the initial experiments did not produce any meaningful results with the data set at hand and consequently we did not use the adaptation to the text length in the submissions and are not reporting the experiments here. The other additional method under investigation was the use of so called blacklists in language identification to which we return in Section 4.2. For all the experiments, as well as for the submission runs, we removed the *NE*.

From the beginning, we suspected that using the in-domain development set would most probably be far more beneficial than using the out-of-domain training set. The language identification methods we were using have a number of parameters which are set using a development set. In light of this, we divided the development set in two parts. We used the first 1,306 lines of Moldavian and the first 1,313 lines of Romanian for the *dev-dev* and the rest was used for *dev-test*.

As the task was very similar to the RDI 2020 task, we continued our experiments from the conclusions we had arrived in 2020 (Jauhiainen et al., 2020). We had experimented with three types of generative classifiers using character *n*-grams: the simple scoring, the sum of relative frequencies and the product of relative frequencies. As the results with the two former were clearly inferior to the product of relative frequencies, we begun our experiments using this naive Bayesian classifier.

4.1 Product of Relative Frequencies

As it had proven to be beneficial on many occasions in the past (Jauhiainen, 2019), we started with a classifier which modified the training and the test data so that all non-alphabetic characters were removed and the remaining alphabetic characters lowercased.

The product of relative frequencies classifier is basically a naive Bayes style generative classifier using the relative frequencies of character *n*-grams as probabilities and is defined as follows in Equation 1:

$$p(g,M) = \begin{cases} \sum_i^{l_{M^F}} -lg_{10}\left(\frac{c(C_g,f_i)}{l_{C_g^F}}\right), \textit{if}\, c(C_g,f_i) > 0 \\ \sum_i^{l_{M^F}} -lg_{10}\left(\frac{1}{l_{C_g^F}}\right) pm, \textit{otherwise} \end{cases} \tag{1}$$

where l_{M^F} is the number of individual features in the mystery text M to be identified and $c(C_g, f_i)$

[2] https://github.com/butnaruandrei/MOROCO

[3] https://github.com/raduionescu/MOROCO-Tweets

is the count of M's ith feature f_i in the training corpus C_g of a dialect g. $l_{C_g^F}$ is the total number of the features of the same type (in this case the same length character n-grams) in C_g.

In case the feature f_i was not found in C_g, a smoothing value was used. As the smoothing value we used the relative frequency of a feature found only once in the training corpus. Instead of multiplying relative frequencies, the sum of their negative logarithms was used. If the smoothing value was was used, the resulting logarithm was multiplied by a penalty modifier, pm, determined using the development set.

First, we set out to optimize the length range of the character n-grams to be used as features as well as the penalty modifier applied to unseen features. In RDI 2020, we ended up using four to five characters as the length with a penalty modifier of 1.40 (Jauhiainen et al., 2020) and these were the parameters we started our experiments with in RDI 2021.

We used the provided training data for training the classifier and the *dev-dev* set to optimize the parameters with a greedy algorithm. Using the greedy algorithm, we found at least a local maximum macro F_1 score of 0.6614 using character n-grams from five to six in length and a penalty modifier of 1.62. This score was very well in line with our best result of 0.6584 in RDI 2020. When evaluated with the *dev-test* set, the same classifier gave a score of **0.6678** which was even better than the one on the data it was optimized for. These results clearly indicated that the first half of the development set was very similar to the second half. Table 1 lists the results of these and some of the following experiments using the training and the development sets.

The second experiment was to combine the training data with the *dev-dev* data, retrain the classifier and test evaluate it using *dev-test*. We used the same parameters and arrived at a macro F_1 score of **0.7203**. This was in line with our suspicions that using the development data itself in this shared task would be crucial in order to obtain high end results.

In the third experiment, we completely left out the training data and used merely the *dev-dev* data to train the classifier. This resulted in a score of **0.7889**, which was again a huge increase on the previous results. This results was now already slightly higher than the winning 0.7876 result of RDI 2020 by Çöltekin (2020). In anticipation of the submission run, we optimized the parameters for the *dev-test* set, ending up using character n-grams from two to six with a penalty modifier of 1.31. Those parameters gave an F_1 score of **0.8072**.

4.2 Blacklist Classifier

The success of using blacklists in language identification of very close languages by Ljubešić et al. (2007) and Tiedemann and Ljubešić (2012) had given us an idea to experiment with combining a blacklist identifier with the generative classifier. We created separate character n-gram lists for each dialect from the training data as well as the *dev-dev* set with the *createmodels.java* program which is part of the HeLI package on GitHub (Jauhiainen et al., 2016).[4]

In the first experiment with blacklist, we used lowercased character n-grams from 4 to 11 in length. We created a blacklist for each of the dialects to be identified. We first populated the blacklist for dialect 1 with those n-grams which were found from the *dev-dev* of dialect 2, but not of the dialect 1. Then, we pruned the list so, that only the n-grams found in the training data of dialect 2 but not in dialect 1 were kept.

We incorporated the blacklists as a preprocessing step on the product of relative frequencies classifier so that the blacklists would judge the mystery text if a blacklisted n-gram would be found from the mystery text.

We continued our practical experiments using the *dev-test* optimized parameters with the NB classifier from the previous experiments in Section 4.1. Adding the blacklists improved the resulting F_1 score slightly to **0.8118**. If the blacklists generated from the *dev-dev* set were not pruned using the training set, the score was merely 0.5719 using the complete blacklist and 0.8076 if only those n-grams that occurred more than 16 times were used.

In the second experiment, we simply combined the *dev-dev* and the training data and used an unpruned blacklist with a frequency cut-off. We evaluated the blacklists with the minimum frequencies from 1 to 10, and the best result of **0.8140** was achieved with the minimum frequency of 7. As the absolute frequency is tightly related to the size of the data set and we were about to use also the *dev-test* set with a submission, we calculated corresponding relative frequencies for each n-gram size.

[4] https://github.com/tosaja/HeLI

Training data	Development data	Testing data	Method	Macro F_1
train	dev-dev	dev-test	NB	0.6678
train + dev-dev	dev-dev	dev-test	NB	0.7203
dev-dev	dev-dev	dev-test	NB	0.7889
dev-dev	dev-test	dev-test	NB	0.8072
dev-dev	dev-test	dev-test	NB + blacklist 1	0.8118
dev-dev	dev-test	dev-test	NB + blacklist 2	0.8140
dev-dev	dev-test	dev-test	NB + blacklist 3	0.8144
dev-dev	dev-test	dev-test	NB + blacklist 3 (all characters)	0.8411
dev-dev	dev-test	dev-test	NB (all characters)	0.8380
dev-dev	dev-test	dev-test	NB + adaptation (all characters)	0.8186

Table 1: Results of the experiments with the training and the development data.

Due to rounding differences, using the relative frequencies gave a score of **0.8144**.

For the third experiment, we extracted the character sets for both dialects from the *dev-dev* set. After some manual inspection, it seemed that there were notable differences in the usage of non-alphabetic characters. In light of this, we modified the identifier to not remove the non-alphabetic characters and we left off the lowercasing as well. We run a new search for the best parameters and received a Macro F_1 score of **0.8411** using character n-grams from two to five, with a penalty modifier 1.61. We decided to use this system for our first submission.

4.3 Language Model Adaptation

For the second submission, we had decided to prepare a version of the naive Bayes classifier using adaptive language models. As our starting point, we reused the system which was used to win the first track of the DMT shared task in 2019 (Jauhiainen et al., 2019a). Combining the blacklist classifier with adaptive language models did not seem straightforward, so we left it for future work.

We first calculated the baseline for the identifier using the same parameters as for the blacklist classifier (character n-grams from 2 to 5 and penalty modifier 1.61) without removing the non-alphabetic characters and conserving the case differences. Using the *dev-dev* for training, the classifier achieved a Macro F_1 score of **0.8380** when tested on *dev-test* without language model adaptation.

In the adaptive version, the classifier keeps a separate record of the lines of the mystery text which have been finally identified. First, every line of the mystery text is identified and the resulting identifications are stored in a temporary table together with the probability differences between the best and the second best language. A larger probability difference corresponds to the identifier having a greater confidence in the identification and are thus called confidence scores. A fixed size fraction of the temporarily identified mystery lines with the highest confidence scores are then processed and their information added to the corresponding language models as well as their identification recorded as finally identified. Then all those lines not yet finally identified are re-identified with the newly adapted language models. This is repeated until all the fractions are processed. The number of fractions the mystery file is divided in is a tunable parameter.

We experimented with several fraction sizes and none really improved the scores when evaluated on the *dev-test* set. This was not totally surprising as we had noted earlier that the *dev-dev* and the *dev-test* sets were very similar. Nevertheless, we decided to submit a run using the adaptive models, as we had no idea how different the actual test set would be from the development set. As a fraction size, we decided to use full division, so that the information from only the most confident line would be added to the language models. This was possible due to the relatively small size of the test set. With the test set size of 5,282 lines the non-adaptive version does 5,282 identifications and the adaptive version with a maximum split 13,949,762 identifications.[5] For the *dev-test* the full split gave a Macro F_1 score of **0.8186**.

5 Submissions and Results

We ended up making two submissions to the shared task. Since it might not be obvious from Section 4 which systems were finally used to generate the

[5] $5,282 * 5,282/2$

submitted results, we will now give short descriptions of these systems.

5.1 Run 1

The results for the first run were produced by using a custom coded language identifier using the product of relative frequencies of character n-grams. Basically it is a naive Bayes classifier using the relative frequencies as probabilities as in Equation 1 (Jauhiainen et al., 2019a). The lengths of the character n-grams used were from 2 to 5. Instead of multiplying the relative frequencies, we summed up their negative logarithms. As a smoothing value, we used the negative logarithm of an n-gram appearing only once multiplied by a penalty modifier. In this case, the penalty modifier was 1.61.

Only the development data was used as training material in this run. All characters were used and not lowercased. The named entity tags were removed in preprocessing from both training and testing data.

In addition to the basic classifier, we used a blacklist of lowercase character n-grams generated from the training and the development data.

5.2 Run 2

The results for the second run were also produced by a language identifier using the product of relative frequencies of character n-grams. The lengths of the character n-grams used were from 2 to 5. Instead of multiplying the relative frequencies we summed up their negative logarithms. As a smoothing value, we used the negative logarithm of an n-gram appearing only once multiplied by a penalty modifier. In this case, the penalty modifier was 1.61.

In addition, we used the same language model adaptation technique as we used with the HeLI method in GDI 2018 (Jauhiainen et al., 2018a). We used one epoch of language model adaptation to the test data.

Only the development data was used as training material in this run. All characters were used and not lowercased. The named entity tags were removed in preprocessing from both training and testing data.

5.3 Results

Table 2 shows the results of the shared task. Our second run won by a considerable margin.

As of this writing, we have no knowledge of the systems used by the other two teams.

Rank	Team	Run	Macro F_1
1	**SUKI**	**2**	**0.7772**
2	UPB	2	0.7325
3	UPB	1	0.7319
4	**SUKI**	**1**	**0.7266**
5	UPB	3	0.6743
6	Phlyers	1	0.6532
7	Phlyers	2	0.5133

Table 2: The results of each team participating on the RDI 2021 shared task.

6 Conclusions

The results would seem to indicate that the tweets in the actual test data were clearly more out-of-domain when compared with the development data than our *dev-dev* set was from the *dev-test* set. The adaptive version, which gave lower scores between *dev-dev* and *dev-test* than the blacklist classifier was clearly superior with the actual test data.

We believe, that if the shared task instructions would have explicitly prohibited using the development data as training material, the results would have been even more interesting. As the development data was available for actually training the classifier as well, it proved to be useful to just forget the actual training data and train the classifier using only the smaller development data available. The best submission of the 2020 edition managed to point this out as it used a far smaller development set as training material, still winning the shared task.

Unfortunately, the two top teams from the RDI 2020 (Çöltekin, 2020; Popa and Ştefănescu, 2020) were not participating in RDI 2021 and the scores of their systems remain a mystery this time. If we compare our RDI 2021 results with the results of the best two teams from 2020, we notice that they are very close.

Acknowledgments

We would like to thank the RDI organizers for organizing the shared task. The research presented in this article has been partly funded by The Finnish Research Impact Foundation, the Kone Foundation, the Academy of Finland, and the University of Helsinki in cooperation with Lingsoft.

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. The COLING 2016 Organizing Committee.

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, Florence, Italy. Association for Computational Linguistics.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

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)*.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Mihaela Gaman, 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 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

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

Tommi Jauhiainen. 2010. Tekstin kielen automaattinen tunnistaminen. Master's thesis, University of Helsinki, Helsinki.

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

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 2018)*, pages 254–262, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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 2018)*, pages 66–75, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2019a. Discriminating between Mandarin Chinese and Swiss-German varieties using adaptive language models. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 178–187, Ann Arbor, Michigan. Association for Computational Linguistics.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2020. 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, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

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. The COLING 2016 Organizing Committee.

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

Martin Kroon, Masha Medvedeva, and Barbara Plank. 2018. When simple n-gram models outperform syntactic approaches: Discriminating between Dutch and Flemish. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 244–253, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nikola Ljubešić, Nives Mikelić, and Damir Boras. 2007. Language Identification: How to Distinguish Similar Languages? In *Proceedings of the 29th International Conference on Information Technology Interfaces (ITI 2007)*, pages 541–546, Cavtat/Dubrovnik, Croatia.

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 (VarDial3), pages 1–14, Osaka, Japan. The COLING 2016 Organizing Committee.

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

Cristian Popa and Vlad Ștefănescu. 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, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Petru Rebeja and Dan Cristea. 2020. A dual-encoding system for dialect classification. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 212–219, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

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

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, Ann Arbor, Michigan. Association for Computational Linguistics.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

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. Association for Computational Linguistics.

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, Dirk Speelman, 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 2018)*, pages 1–17, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Yves Scherrer, Tanja Samardžić, Francis Tyers, Miikka Silfverberg, Natalia Klyueva, Tung-Le Pan, Chu-Ren Huang, Radu Tudor Ionescu, Andrei M. 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*, pages 1–16, Ann Arbor, Michigan. Association for Computational Linguistics.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A report on the DSL shared task 2014. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 58–67, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

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*, pages 1–9, Hissar, Bulgaria. Association for Computational Linguistics.