Comparing Approaches to Dravidian Language Identification

Tommi Jauhiainen[1,3], Tharindu Ranasinghe[2], Marcos Zampieri[3]
[1]University of Helsinki, Finland
[2]University of Wolverhampton
[3]Rochester Institute of Technology, USA
tommi.jauhiainen@helsinki.fi

Abstract

This paper describes the submissions by team HWR to the Dravidian Language Identification (DLI) shared task organized at VarDial 2021 workshop. The DLI training set includes 16,674 YouTube comments written in Roman script containing code-mixed text with English and one of the three South Dravidian languages: Kannada, Malayalam, and Tamil. We submitted results generated using two models, a Naive Bayes classifier with adaptive language models, which has shown to obtain competitive performance in many language and dialect identification tasks, and a transformer-based model which is widely regarded as the state-of-the-art in a number of NLP tasks. Our first submission was sent in the closed submission track using only the training set provided by the shared task organisers, whereas the second submission is considered to be open as it used a pretrained model trained with external data. Our team attained shared second position in the shared task with the submission based on Naive Bayes. Our results reinforce the idea that deep learning methods are not as competitive in language identification related tasks as they are in many other text classification tasks.

1 Introduction

Discriminating between similar languages (e.g. Bulgarian and Macedonian or Croatian and Serbian), language varieties (e.g. Brazilian and European Portuguese), and dialects is one of the main challenges in automatic language identification (LI) in texts. This issue has been addressed in a couple recent surveys (Jauhiainen et al., 2019c; Zampieri et al., 2020) and evaluation papers (Goutte et al., 2016). Furthermore, it has been the topic of a number of competitions such as TweetLID (Zubiaga et al., 2016) and many shared tasks organized in past VarDial workshops (Zampieri et al., 2018, 2019; Găman et al., 2020).

In this paper, we revisit the challenge of discriminating between similar languages on a challenging code-mixed data set containing three South Dravidian Languages: Kannada, Malayalam, and Tamil. In addition, there were code-mixed comments on other languages as well. The data set contains 22,164 YouTube comments divided into 16,674 instances for training and 4,590 instances for testing, each containing a mix of languages. The data set was released by the organizers of the Dravidian Language Identification (DLI) shared task part of the VarDial Evaluation Campaign 2020 (Chakravarthi et al., 2021).[1] The goal of the task is to train computational models able to identify the language of each comment in a test set presented in Section 3.

The DLI debuted as the first competition on similar language identification using code-mixed data associated with the VarDial workshops. Very similar multilingual language identification shared tasks were organized as part of Forum for Information Retrieval Evaluation (FIRE) meetings in 2013 and 2014 (Roy et al., 2013; Choudhury et al., 2014).[2] A similar competition to the DLI are also the shared tasks on Language Identification in Code-Switched Data (Solorio et al., 2014; Molina et al., 2016). These shared tasks addressed intra-sentential language identification with word level labeling whereas the DLI shared task is language identification shared task with predictions at the document level. We take this opportunity to evaluate the performance of two models for this task, a Naive Bayes classifier using adaptive language models and a transformers-based system described in detail in Section 4.

[1] https://sites.google.com/view/vardial2021/evaluation-campaign
[2] http://fire.irsi.res.in/fire/2021/home

Proceedings of the 8th VarDial Workshop on NLP for Similar Languages, Varieties and Dialects, pages 120–127

2 Related Work

The task of automatic language identification of texts has been under continuous research since 1960s as is witnessed, for example, by the works of Mustonen (1965), House and Neuburg (1977), Henrich (1989), Grefenstette (1995), and Martins and Silva (2005). The problem of discriminating between similar languages, language varieties, and dialects is a particularly challenging one and it has also been addressed by a number of studies such as Tiedemann and Ljubešić (2012); Tan et al. (2014); Malmasi and Zampieri (2017b,a), and the aforementioned shared tasks at the VarDial workshop. In addition to addressing the issue of discriminating between similar languages, Jauhiainen et al. (2019c) provide an extensive overview of the history and methods used in language identification of texts in general.

2.1 Language Identification of South Dravidian Languages

Even though the DLI 2021 shared task was the first time a shared task solely focused on discriminating between Dravidian languages, Dravidian languages have been part of language repertoire of LI research before. Most of the research so far has focused on texts written with non-Roman script. Notable exception to the trend is the first subtask of the Transliterated Search shared which was organized as part of FIRE 2014 (Choudhury et al., 2014), where the goal was to label individual transliterated words in code-mixed search queries. The training set of the subtask included code-mixed sentences in English together with one of the Indian languages: Bangla, Gujarati, Hindi, or Malayalam. In addition, the test set included code-mixed sentences with Kannada and Tamil as unseen languages. Choudhury et al. (2014) list three teams which submitted results for the Dravidian languages, but to the best of our knowledge there are no system description papers available for these submissions.

Using Multivariate Analysis (MVA) and Principal Component Analysis (PCA), Vinosh Babu and Baskaran (2005) attained a 100% performance in language identification between six Tamil dialects (Iscii, Shree-Tam, Tab, Tam, Tscii, and Vikatan). The exact test sizes they used are not known, but they considered 500 bytes "too small", which indicates that the texts they processed were generally much longer than those part of the DLI shared task test set.

Murthy and Kumar (2006) focused on smaller text samples in pair wise language identification of several Indian languages, including those of Tamil, Malayalam, and Kannada with Multiple Linear Regression (MLR). They used the number of aksharas (a sort of syllables generally used in Indian scripts) as the measure of the length of text and obtained F-scores of over 0.99 with texts of only 10 aksharas in length when discriminating between two of the South Dravidian languages. In their research, they show that features based on aksharas (*e.g.* akshara *n*-grams) work better than features based on bytes.

Goswami et al. (2020) experiment with supervised and unsupervised methods in dialect identification using, among others, a Dravidian data set containing Tamil, Telugu, Malayalam, and Kannada.

The three South Dravidian languages were a part of larger repertoire of the language identifiers presented by, for example, Majliš (2011) and Kocmi and Bojar (2017). Hanumathappa and Reddy (2012) conducted identification experiments between Kannada and Telugu, a Central Dravidian language.

3 Shared Task Setup and Data

The evaluation measure in the DLI shared task was the macro F_1 score, which gives equal value for each language independent of their actual distribution in the test set.

The data set provided by the DLI organizers contains a total of 22,164 YouTube comments written in a mix of English and one of the aforementioned South Dravidian languages (Chakravarthi et al., 2020a,b; Hande et al., 2020). In addition to the target languages, the data included comments in other languages as well. It was divided into 16,674 instances for training and 4,590 instances for testing. The number of instances for each language is show in Figure 1.

Set	kan	mal	tam	other	Total
Training	493	4,204	10,969	1,008	16,674
Test					4,590
Total					22,164

Table 1: Number of instances in the DLI dataset for Kannada (kan), Malayalam (mal), and Tamil (tam).

In order to evaluate and compare our methods using the training data, we divided the training data into training and development portions. For training, we

used the first 90% of comments for each language and the rest was set aside for development. This way, we retained the original distribution of different labels as the provided training data seemed not to be in a random order. For example, the last 10% of the training comments did not include comments in Malayalam at all.

For the open track, we considered using corpora collected for the use of the SUKI-project (Jauhiainen et al., 2015a), but quickly found that for example the SUKI data for Tamil consisted of texts written completely using the Devanagari script. As the YouTube comments used in the shared task were mostly written in Latin alphabet, we did not pursue using external corpora further.

4 Methods and Experiments with the development data

In our development environment, we experimented with several different methods: simple scoring, sum of relative frequencies, the product of relative frequencies (Naive Bayes, NB), NB with language model adaptation, HeLI, language set identification, and transformers.

4.1 Simple scoring, sum of relative frequencies, and Naive Bayes

None of the three methods are equipped with any special ways of handling multilingual texts and simply act in a similar fashion as each line would actually be monolingual. The possible multilinguality of the input texts is handled by the language models derived from the training material actually representing a mixture of languages.

The implementation of these methods was the same, which was used by Jauhiainen et al. (2019a) in the Discriminating between Mainland and Taiwan variation of Mandarin Chinese (DMT) shared task. We have now published this software titled "Tunnista" in GitHub with an MIT license.[3] The software is used to automatically evaluate the performance of the methods using different ranges of character *n*-grams and penalty modifiers.

In the evaluations of all three methods, we removed all non-alphabetical characters from the training and the test material and also lowercased all the remaining characters.

Simple scoring was evaluated with all possible character *n*-gram ranges from 1 to 10 and from 6 to 11. The best micro-F_1 score of 0.9225 was attained using character *n*-grams from 7 to 8 and the best macro-F_1 of **0.7321** with character *n*-grams from 7 to 10. The sum of relative frequencies was evaluated with all possible character *n*-gram ranges from 1 to 11. The best micro-F_1 score of 0.8048 was attained using character *n*-grams from 3 to 9 and the best macro-F_1 of **0.6897** with character *n*-grams from 7 to 9. It is noticeable how much closer the macro F_1 scores are to each other when compared with the micro F_1 scores.

The NB classifier was evaluated with several combinations of character *n*-grams and penalty modifiers ranging from *n*-gram length of 1 to 11 and penalty modifiers between 1.2 and 2.5. The classifier obtained its best micro F_1 score of 0.9339 using character *n*-grams from 1 or 2 to 6 with penalty modifiers ranging from 2.37 to 2.42. The best macro F_1 score, **0.8609** was attained using character *n*-grams from 2 to 6 with penalty modifiers ranging from 2.14 to 2.16. The NB classifier was clearly the best of the three methods provided by the "Tunnista" program.

4.2 Product of Relative Frequencies with Adaptive Language Models

As the macro F_1 score was used as the evaluation measure of the shared task, we used the best parameters (character *n*-grams from 2 to 6, with penalty modifier of 2.15) from the previous experiments as the basis for our experiments with the adaptive version of the naive Bayes classifier.

The adaptation method uses several parameters which have to be optimized using the training and the development material. The first parameter is the number of splits the whole material to be identified is divided in. The actual division into splits happens after each time the test set is preliminarily identified and ordered so that the mystery texts with the highest difference between the log probabilities of the most probable and the second most probable language are on the top of the list. When incorporating new information from the text to be identified, the highest split is processed first. After its information has been added to the language models, all the remaining mystery texts are again preliminarily identified and divided into same sized splits. Again the information from the best split is incorporated into language models and so on, until all the splits have been processed.

We evaluated different split sizes between zero and the so called full split, which means the number

[3] https://github.com/tosaja/Tunnista

of splits was equal to the number of mystery texts in the test set. Table 2 shows the development of the F_1 score relative to the number of splits k.

k	macro F_1-score
1	0.8609
2	0.8595
4	0.8623
10	0.8648
20, 40, 100, 200, 400, 800, max	**0.8663**

Table 2: The macro F1-scores obtained by the NB identifier using adaptation with different values of k when evaluated on the development partition of the DLI training set.

We also use the log probability difference between the best and second best scores as a confidence score. In case confidence threshold, CT, is used, no information from mystery lines with confidence score equal or below CT is incorporated into the language models. We evaluated several CT values with k of 20, but the best results were obtained without using the confidence threshold at all.

4.3 HeLI 2.0 method

The third series of experiments was conducted using a language identifier based on the HeLI method (Jauhiainen et al., 2016). The actual implementation used was that of the HeLI 2.0 method (Jauhiainen et al., 2019b) with adaptive language models from the GDI 2019 shared task (Jauhiainen et al., 2019a).[4] We used the $createmodels.java$[5] program to generate language models of words and character *n*-grams of up to 12 characters in length, both lowercased and with the original casing. All non-alphabetical characters were also removed.

Table 3 lists some combinations of parameters and the results they achieved for the development data in our experiments. Column heading lnr refers to the length range of lowercased character *n*-grams, onr to those with original casing, lw and ow tell whether lowercased or origical words were used (y) or not (n). Column heading pm refers to the used penalty modifier.

From these experiments, it was clear that the naive Bayes based language identifier fared better than the one implementing the HeLI method. We also experimented with HeLI 2.0 using language model adaptation scheme identical to what we described in Section 4.2. These experiments failed to increase the macro F_1 score at all.

lnr	*onr*	*lw*	*ow*	*pm*	macro F1-score
1-11	1-11	y	y	1.17	0.8334
1-11	1-11	y	n	1.14	0.8305
1-11	-	y	y	1.18	0.8308
-	1-11	y	y	1.17	0.8334
1-10	1-10	y	y	1.17	0.8334
1-9	1-9	y	y	1.17	0.8334
1-8	1-8	y	y	1.17	0.8334
1-7	1-7	y	y	1.10	0.8338
1-6	1-6	y	y	1.11	**0.8403**
1-5	1-5	y	y	1.09	0.8369
2-6	2-6	y	y	1.11	**0.8403**
3-6	3-6	y	y	1.10	0.8396
2-6	2-6	y	n	1.10	0.8319
2-6	2-6	n	y	1.11	**0.8403**

Table 3: The macro F1-scores obtained by HeLI-based identifier when evaluated on the development partition of the DLI training set. See Section 4.3 for the explanations of the column headings. The best results are in bold.

4.4 Language Set Identification

As the mystery lines in the DLI data set were actually multilingual, we wanted to experiment with a system capable of detecting several languages in one text. We set out to incorporate the language set identification method deviced by Jauhiainen et al. (2015b) into the naive Bayes identifier with adaptive language models. A server version of the language set identifier using the HeLI method is currently available at GitHub.[6] It has been successfully used as part of corpus collection and creation pipeline resulting to the Wanca corpora, which has been used in the Uralic Language Identification (ULI) shared task (Jauhiainen et al., 2020a,b). We managed to combine the method to the naive Bayes classifier and run some initial experiments, in which using the language set identification did not improve the results. Unfortunately, we did not have time to finalize our experiments as we were left contemplating about the nature of the multilinguality inherent in the DLI data set and were not encouraged by our initial results.

4.5 Transformers

The system for our second submission (described in Section 5.2) was based on pretrained transformer

[4]The publication of the HeLI 2.0 implementation is still currently in our queue.

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

[6]https://github.com/tosaja/TunnistinPalveluMulti

models: multilingual BERT (Devlin et al., 2019) and XLM-RoBERTa (XLM-R) (Conneau et al., 2019). The system used pretrained language models available from the Hugging Face Team [7]. Transformer are effective than RNN based deep learning architectures (Hettiarachchi and Ranasinghe, 2019) in text classification tasks (Ranasinghe et al., 2019; Ranasinghe and Hettiarachchi, 2020; Pitenis et al., 2020).

It was also evaluated using the same development set which was used in the previously described experiments. It the initial experiments it achieved a macro F_1 score of 0.785, which was considerably lower than the 0.861 gained by the simple naive Bayes model even though it used pretrained models as opposed to the naive Bayes which was using only the data provided for the DLI task. We considered generating additional training material from the SUKI data and other available Dravidian corpora written with the native script using automatic transliteration.[8] However, we were doubtful about the performance of such libraries since the texts they were trained with were not natural romanized writing as opposed to the YouTube comments part of the DLI shared task and did not pursue this further.

5 Submissions and Results

In this Section, we summarize the systems we used in our submission and provide the results obtained on the test set.

5.1 System 1: Bayesian Classifier

The results of our first submission were produced using the adaptive version of the naive Bayes classifier with the best parameters attained in the experiments detailed in Sections 4.1 and 4.2.

The system is based on a naive Bayes classifier using the relative frequencies of character *n*-grams as probabilities (Jauhiainen et al., 2019a). Non-alphabetic characters were removed and the rest were lowercased. The lengths of the character *n*-grams used were from two to six. 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 2.15.

In addition, we used identical language model adaptation technique as was used with the HeLI method in GDI 2018 (Jauhiainen et al., 2018). We used one epoch of language model adaptation to the test data. The *n*-gram models used, the penalty modifier, the confidence value, and the number of splits in adaptation were optimized using 10% of the training data as the development data.

5.2 System 2: Transformers

These results were produced by using a pretrained transformer model which has been used in a number of NLP tasks like text classification (Pitenis et al., 2020), span classification (Ranasinghe and Zampieri, 2021), word similarity (Hettiarachchi and Ranasinghe, 2020a), question answering (Yang et al., 2019) etc. We pass the sentence through the transformer model and add a softmax layer on top of the [CLS] token as a normal classification architecture with transformers (Ranasinghe and Hettiarachchi, 2020). We fine-tune all the parameters from transformer model as well as the softmax layer jointly by maximising the log-probability of the correct label. This architecture has been used widely in many text classification tasks (Ranasinghe et al., 2019; Ranasinghe and Zampieri, 2020) that includes Malayalam code-mix texts too (Ranasinghe et al., 2020). We did not perform any preprocessing to this architecture. Considering the pretrained transformer models that supports Kannada, Malayalam and Tamil we used multilingual bert (Devlin et al., 2019) and XLM-R large (Conneau et al., 2019) models.

We divided the dataset into a training set and a validation set using 0.8:0.2 split on the data set. We predominantly fine tuned the learning rate and number of epochs of the classification model manually to obtain the best results for the validation set. We obtained $1e^{-}5$ as the best value for learning rate and 3 as the best value for number of epochs. We first fine-tune multiple transformer models with different random seeds. For each input, we output the best predictions made by the transformer along with the probability and sum up the probability of the predictions from each model together. The output of the ensemble model is the prediction with the highest probability. This voted ensemble method has improved results in many tasks (Hettiarachchi and Ranasinghe, 2020b) for transformers.

[7] https://huggingface.co/transformers/pretrained_models.html

[8] Indic transliteration tools provide such functionalities - https://pypi.org/project/indic-transliteration/

5.3 Results on the Test Set

Table 4 shows the results of the shared task. Our first run was clearly better than the second and not far behind the results of the LAST-team.

Rank	Team	Run	Macro F_1
1	LAST	1	0.93
	LAST	2	0.92
	LAST	3	0.92
2	**HWR**	**1**	**0.92**
2	NYAEL	1	0.92
	NAYEL	2	0.91
4	Phlyers	1	0.89
	Phlyers	2	0.89
	HWR	**2**	**0.89**
	NAYEL	3	0.84

Table 4: The results of each team participating on the DLI shared task in terms of Macro F1. Our results are displayed in bold.

6 Discussion

We did not experiment with the NB identifier using non-alphabetic characters nor with the original casing of the alphabetic characters. In light of the results for the HeLI method in Table 3, it might be a promising direction. It seems that at least for the HeLI method, the original character *n*-grams were more important than lowercased as can be seen from the third and fourth rows of the table. The same NB method was used also in the winning submission of the Romanian Dialect Identification (RDI 2021) shared task organized as part of the same campaign as DLI (Chakravarthi et al., 2021). In RDI, the use of non-alphabetic characters and original casing proved out to be quite important (Jauhiainen et al., 2021) and might have very well cost us the small difference between the second and the first positions in the DLI shared task.

Even though the difference in performance between the NB model and the transformers was only 3 percentage points in the test set, the fact that the transformers did not outperform the simple naive Bayes classifier deserves special attention. One of the reasons to the inferior performance of the pretrained models is probably that the comments contained code-mixed sentences kind of which the pretrained language models like BERT and XLM-R had not seen before.

7 Conclusion

We present the submissions by team HWR to the Dravidian Language Identification (DLI) shared task at VarDial 2021. The DLI shared task featured a challenging dataset including YouTube comments written in Roman script containing code-mixed text with English and one of the three South Dravidian languages: Kannada, Malayalam, and Tamil. Our two systems, obtained competitive performance with the NB system achieving 2nd position in the competition.

Our results are in line with the general trend of deep learning methods not being overtly competitive in language identification tasks as discussed in Medvedeva et al. (2017).

Acknowledgments

We would like to thank the DLI organizers for making this interesting dataset available.

This research has been partly funded by The Finnish Research Impact Foundation in cooperation with Lingsoft.

References

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. In *Proceedings of SLTU/CCURL*.

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 SLTU/CCURL*.

Monojit Choudhury, Gokul Chittaranjan, Parth Gupta, and Amitava Das. 2014. Overview of FIRE 2014 Track on Transliterated Search. In *Proceedings of FIRE*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Koustava Goswami, Rajdeep Sarkar, Bharathi Raja Chakravarthi, Theodorus Fransen, and John Philip McCrae. 2020. Unsupervised deep language and dialect identification for short texts. In *Proceedings of COLING*.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating Similar Languages: Evaluations and Explorations. In *Proceedings of LREC*.

Gregory Grefenstette. 1995. Comparing Two Language Identification Schemes. In *Proceedings of JADT*.

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 VarDial*.

Adeep Hande, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2020. KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection. In *Proceedings of PEOPLES*.

M. Hanumathappa and Mallamma V. Reddy. 2012. Natural Language Identification and Translation Tool for Natural Language Processing. *International Journal of Science and Applied Information Technology*, 1(4).

Peter Henrich. 1989. Language Identification for the Automatic Grapheme-to-phoneme Conversion of Foreign Words in a German Text-to-speech System. In *Proceedings of EUROSPEECH*.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2019. Emoji powered capsule network to detect type and target of offensive posts in social media. In *Proceedings of RANLP*.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2020a. BRUMS at SemEval-2020 task 3: Contextualised embeddings for predicting the (graded) effect of context in word similarity. In *Proceedings of SemEval*.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2020b. InfoMiner at WNUT-2020 task 2: Transformer-based covid-19 informative tweet extraction. In *Proceedings of W-NUT*.

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.

Heidi Jauhiainen, Tommi Jauhiainen, and Krister Lindén. 2015a. The Finno-Ugric Languages and The Internet Project. In *Proceedings of IWCLUL*.

Heidi Jauhiainen, Tommi Jauhiainen, and Krister Lindén. 2020a. Building web corpora for minority languages. In *Proceedings of WAC*.

Tommi Jauhiainen, Heidi Jauhiainen, and Lindén. 2021. Naive Bayes-based Experiments in Romanian Dialect Identification. In *Proceedings of VarDial*.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2018. HeLI-based experiments in Swiss German dialect identification. In *Proceedings of VarDial*.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2019a. Discriminating between Mandarin Chinese and Swiss-German varieties using adaptive language models. In *Proceedings of VarDial*.

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 VarDial*.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2015b. Language Set Identification in Noisy Synthetic Multilingual Documents. In *Proceedings of CICLing*.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2016. HeLI, a Word-Based Backoff Method for Language Identification. In *Proceedings of VarDial*.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2019b. 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. 2019c. Automatic Language Identification in Texts: A Survey. *Journal of Artificial Intelligence Research*, 65:675–782.

Tom Kocmi and Ondřej Bojar. 2017. LanideNN: Multilingual Language Identification on Character Window. In *Proceedings of EACL*.

Martin Majliš. 2011. Large Multilingual Corpus. Master's thesis, Charles University in Prague, Prague.

Shervin Malmasi and Marcos Zampieri. 2017a. Arabic Dialect Identification Using iVectors and ASR Transcripts. In *Proceedings of VarDial*.

Shervin Malmasi and Marcos Zampieri. 2017b. German Dialect Identification in Interview Transcriptions. In *Proceedings of VarDial*.

Bruno Martins and Mário J. Silva. 2005. Language Identification in Web Pages. In *Proceedings of SAC*.

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 VarDial*.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of CodeSwitch*.

Kavi Narayana Murthy and G. Bharadwaja Kumar. 2006. Language Identification from Small Text Samples. *Journal of Quantitative Linguistics*, 13(1):57–80.

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

Zeses Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020. Offensive Language Identification in Greek. In *Proceedings of LREC*.

Tharindu Ranasinghe, Sarthak Gupte, Marcos Zampieri, and Ifeoma Nwogu. 2020. WLV-RIT at HASOC-Dravidian-CodeMix-FIRE2020: Offensive Language Identification in Code-switched YouTube Comments. In *Proceedings of FIRE*.

Tharindu Ranasinghe and Hansi Hettiarachchi. 2020. BRUMS at SemEval-2020 task 12: Transformer based multilingual offensive language identification in social media. In *Proceedings of SemEval*.

Tharindu Ranasinghe and Marcos Zampieri. 2020. Multilingual offensive language identification with cross-lingual embeddings. In *Proceedings of EMNLP*.

Tharindu Ranasinghe and Marcos Zampieri. 2021. MUDES: Multilingual Detection of Offensive Spans. In *arXiv preprint arXiv:2102.09665*.

Tharindu Ranasinghe, Marcos Zampieri, and Hansi Hettiarachchi. 2019. BRUMS at HASOC 2019: Deep Learning Models for Multilingual Hate Speech and Offensive Language Identification. In *Proceedings of FIRE*.

Rishiraj Saha Roy, Monojit Choudhury, Prasenjit Majumder, and Komal Agarwal. 2013. Overview of the FIRE 2013 Track on Transliterated Search. In *Proceedings of FIRE*.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of CodeSwitch*.

Liling Tan, Marcos Zampieri, Nikola Ljubešic, and Jörg Tiedemann. 2014. Merging Comparable Data Sources for the Discrimination of Similar Languages: The DSL Corpus Collection. In *Proceedings of BUCC*.

Jörg Tiedemann and Nikola Ljubešić. 2012. Efficient Discrimination between Closely Related Languages. In *Proceedings of COLING*.

J. Vinosh Babu and S. Baskaran. 2005. Automatic Language Identification Using Multivariate Analysis. In *Proceedings of CICLing*.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of NAACL*.

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 VarDial*.

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 VarDial*.

Marcos Zampieri, Preslav Nakov, and Yves Scherrer. 2020. Natural Language Processing for Similar Languages, Varieties, and Dialects: A Survey. *Natural Language Engineering*, 26:595–612.

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.