Early Guessing for Dialect Identification

Anonymous ACL submission

Abstract

This paper deals with the problem of incremental dialect identification. Our goal is to reliably determine the dialect before the full utterance is given as input. The major part of the previous research on dialect identification has been model-centric with a focus on performance. We address a new question: How much input is needed to identify a dialect? Our approach is a data-centric analysis that results in general criteria for finding the shortest input needed to make a plausible guess. Working with two sets of dialects (Swiss German and Indo-Aryan languages), we show that the dialect can be identified well before the end of the input utterance. To determine the optimal point for making the first guess, we propose a heuristic that involves calibrated model confidence (temperature scaling) and input length. We show that the same input shortening criteria apply to both of our data sets. While the performance with the early guesses is still below the performance on the full input, the gap is smaller when the overall performance of the fine-tuned model is better[1].

1 Introduction

Language identification depends very much on what kind of languages we are discriminating. If languages to be discriminated are distant (e.g. Russian vs. Chinese), the task is very easy and a short sequence of words provides enough information to assign the correct class. But if languages are similar and written in the same script (e.g. Russian vs. Ukrainian), much longer samples are needed to encounter the discriminating features (Tiedemann and Ljubešić, 2012). The task is even harder when dealing with non-standard orthography, which we find in written dialects and user posts on the internet (Zampieri et al., 2017).

Current research is mostly concerned with improving the performance on the task by applying increasingly sophisticated methods, including pre-trained models, whose usefulness is still not fully confirmed (Jauhiainen et al., 2021). However, many other aspects of the task may play an important role in practical applications. One of such challenges are the possibility to make early guesses on the language or dialect, before seeing the whole message. Such a feature can be especially useful for more dynamic classification of a continuous stream of messages to choose most suitable methods for end-user tasks.

In this paper, we address the problem of early guessing in dialect identification mostly from the data-centric point of view, but considering some model-centric issues too. We pose the following research question:

RQ: Given the input text and an existing pre-trained model, is it possible to achieve the same or similar performance by observing a prefix of an utterance compared to the full utterance?

To answer this question, we search for general criteria for shortening the input so that the model performance is the same or similar to the performance obtained with the full input. We perform experimental studies in two settings: dialect identification with non-standard writing and language identification for similar languages with standard writing. We show that the same shortening criteria apply to both settings and that the early guessing performance depends on the overall performance of the model.

2 Related Work

The task of dialect identification and discrimination between similar languages is mostly addressed in the scope of the VarDial Evaluation Campaign (Zampieri et al., 2017, 2018, 2019). The organisers of the tasks released datasets for various cases of dialects and similar languages, such as Swiss-German, Indo-Aryan, Uralic, Romanian, Arabic, Slavic, Chinese, etc. Competing teams proposed

[1] We plan to release the code for replicating the analyses.

	GDI	ILI
Train	14647	68453
Dev	4659	8286
Test	4752	9032

Table 1: The size of datasets (expressed as the number of utterances). GDI : German Dialect Identification. ILI: Indo-Aryan Language Identification.

various solutions including n-gram features and tf-idf features using standard machine learning classifiers such as SVM and Naive Bayes, but also deep learning approaches using word2vec, LSTMS, CNN's, RNN's, etc. (Ali, 2018; Ciobanu et al., 2018b; Jauhiainen et al., 2018; Gupta et al., 2018; Çöltekin et al., 2018; Ciobanu et al., 2018a; Bernier-Colborne et al., 2021). With the advent of transformer-based models, we see wide use of pre-trained models in dialect classifications (Popa and Ștefănescu, 2020; Zaharia et al., 2020; Ljubešić and Lauc, 2021), but traditional approaches based on n-gram statistics still seem to be most successful on this task.

The research in dialect classification is mainly directed towards improving the model performances using various architectures. Usually, language and dialect identifications tasks are carried out in a supervised setup, but with little data analysis. In contrast to the previous work, the main focus of our work is not on improving the performance, but on achieving good performance with minimal input. While good models are always desired, a data-centric exploration that we propose is needed to better exploit the existing classifiers in practical applications.

3 Data

For our experiments, we select two datasets offered by the VarDial Evaluation Campaign (see Section 2): German Dialect Identification (GDI)[2] and Indo-Aryan Language Identification (ILI).[3]. Swiss German dialect/ GDI dataset represents four areas: Basel, Bern, Lucerne, and Zurich. Training and the test datasets are obtained from the ArchiMob corpus of Spoken Swiss German with 43 oral history interviews (Samardzic et al., 2016). GDI datasets are available from the years 2017-2019. In GDI-2018 data, a fifth "surprise dialect" (Valais Swiss German dialect) was introduced in the test set. The participants could take part either in four-way classification (without the surprise dialect) or in the five-way classification. We work mostly with the GDI-2018, but in the 4-way classification setting. The ILI task is about identifying five closely-related languages from the Indo-Aryan language family, namely, Hindi (also known as Khari Boli), Braj Bhasha, Awadhi, Bhojpuri, and Magahi. For each language, 15,000 sentences are extracted mainly from the literature domain. The sources were previously published either on the internet or in print. These languages are often mistakenly considered to be varieties of Hindi. Table 1 reports the data statistics of GDI and ILI datasets.

4 Methods

We perform incremental analysis by running the same classifier on varied substrings of the test input. We start with the first word, then repeat the classification with the first two words and so on until we reach the end of the utterances. We refer to all the incremental substrings as fragments. We observe the performance of the model at each incremental step and analyze its state (confidence) to determine the earliest point when a plausible guess can be made. We perform extensive analysis on the influence of different parameters that directly and indirectly affect the model performance after applying the shortening criteria.

Models We used the state-of-the-art pre-trained BERT-based models, which had given high performance on similar tasks. For the GDI data set, we compared three models: BERT-base-cased model(Devlin et al., 2019), multilingual BERT (mBERT) and German BERT[4]. In the case of the ILI dataset we compared four models: BERT-base-cased, mBERT, IndicTransformers (Jain et al., 2020) and IndicBERT(Kunchukuttan et al., 2020). IndicBERT covers 12 languages including Hindi, Assamese, Tamil, English, Gujarathi, Malayalam etc., trained using AI4Bharat's [5] monolingual corpus and is based on multilingual ALBERT. IndicTransformers[6] is a BERT model trained with 3 GB of monolingual data from OSCAR corpus [7] and covers three languages, viz., Hindi, Bengali and Telugu.

[2] https://drive.switch.ch/index.php/s/DZycFA9DPC8FgD9

[3] https://github.com/kmi-linguistics/vardial2018

[4] https://www.deepset.ai/german-bert

[5] https://indicnlp.ai4bharat.org/indic-bert/

[6] https://huggingface.co/neuralspace-reverie

[7] https://oscar-corpus.com/

Input shortening We first tokenize the input sentence by splitting on white spaces. We then create fragments that consist of incrementally increased prefixes of the original utterance. The length of fragments ranges between 1 and N, where N is the length (in tokens) of the original utterance. For example, consider the test sentence: *'das haisst im klarteggst'* of length *N*=4. The incremental fragments will be:

[*'das'*
'das haisst'
'das haisst im'
'das haisst im klarteggst']

This process gives 42797 fragments for the 4752 test cases in the GDI dataset. In ILI, we obtain 170710 fragments from 9032 test cases. For each fragment, we obtain predictions using the same fine-tuned model. We collect the information about model prediction and its confidence for further analyses.

Upper Bound To see whether correct predictions are possible before seeing the full utterance, we first find the minimum length fragment at which a correct prediction is made. For instance, length four (the fourth line in the example above) will be selected as the optimal shortened input for the given utterance since the predicted class is wrong in the previous three fragments (lines 1-3 in the example above). In this case, length 4 is the shortest length at which the correct prediction is obtained. We find such fragments for each original test utterance (one fragment per utterance) and then compute the classification accuracy with respect to these optimal input lengths.

Measured in this way, the accuracy scores are higher compared to the full-input classification. In the case of GDI, we get 80% (compared to 62% on the full input). For ILI, we obtained an upper bound of 94% compared to the 90% accuracy exhibited by the best baseline model. We consider this accuracy to be our upper bound: this is what could be achieved if we knew where to cut the input utterance in each case. This provides us with an empirical justification for the goal of our study, which is finding criteria for shortening the input. The general idea is that dialects can be identified within a range of length of input *n*, where *n*<*N*, *N* is the length of the original utterance.

Length analysis The first method that we apply to find the optimal input shortening point is an analysis of the relationship between the lengths of fragments and the accuracy obtained from the model. We consider the accuracy of predictions at all fragment lengths to find out whether there is any specific length point at which we can shorten the inputs to obtain correct predictions consistently. The results of this analysis are presented in Section 5. Our explorations pointed out that there is no such a length point in absolute terms, but that length is an important parameter to be considered for devising the final criteria.

Model confidence analysis with Temperature Scaling This method relies on the fact that the model is not equally confident about all outputs predictions. We thus extract confidence scores for each prediction in order to find out whether this information may facilitate finding the optimal point for input shortening. Extracting the information about the model's confidence raises the question of how well this information can be trusted. The confidence scores of the model can very high (close to 1) even when the predictions are incorrect. Calibration is a method to disincentivize a model from being over-confident. Although the transformer models are considered to be well-calibrated (Desai and Durrett, 2020), methods such as temperature scaling (Guo et al., 2017) and label smoothing(Müller et al., 2019) can improve the calibration. We expect this help especially for the case of GDI data, where the overall performance is rather low compared to the other datasets.

We explore temperature scaling to calibrate the prediction probabilities of our model: we divide the non-normalized logits (before the softmax operation) with the scalar temperature hyperparameter T. After this step, the prediction probability is obtained using the usual Softmax function. The values of the parameter $T > 0$ is the same for all classes and it is optimized with respect to the Negative-Log-Likelihood (NLL) loss on the validation set. To compare the models after and before calibration we use Expected Calibration Error (ECE) as shown in Equation (1).

$$ECE = \sum_{k=1}^{K} \frac{b_k}{n} |acc(k) - conf(k)| \qquad (1)$$

Calibration is formally expressed as a joint distribution which can be approximated by binning the predictions to K disjoint sets. Each bin will have b_k predictions and n is the number of samples. ECE is defined as the weighted average of the difference

Dataset	Model	Full	Short
GDI	BERT-base-cased	**62**	55.2
	mBERT	59	50.8
	German BERT	60	–
ILI	BERT-base-cased	81	56.5
	mBERT	88	69.9
	IndicBERT	84	–
	IndicTransformers	**90**	73.7

Table 2: The accuracy (%) with different pretrained models on full utterances and on shortened input.

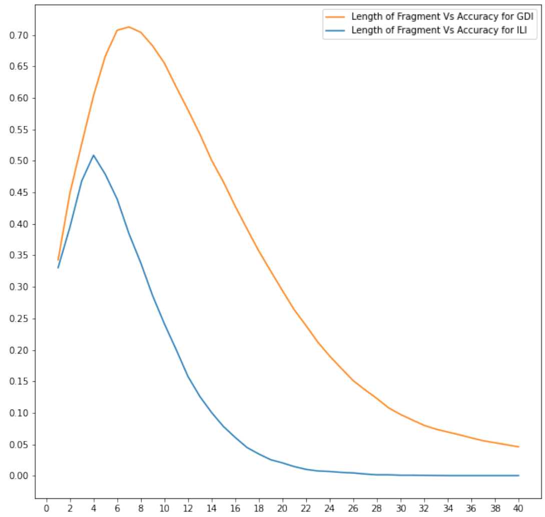

Figure 1: Accuracy Related to Each Fragment Length for GDI and ILI Datasets

between each bin's accuracy and confidence or posterior probability. A perfectly calibrated model has *conf(k)* = *acc(k)* for each bucket of real-valued predictions.

5 Experiments and Results

Each model was trained for 4 epochs with Adam optimizer using a learning rate of 2e-5 on the corresponding training set using 1 Tesla K80 GPU. We used the pre-trained models from the HuggingFace library [8]. Table 2 shows the classification accuracy with full input and with shortened input. Choosing a pre-trained model based on a close language turned out to be important only for the ILI dataset, while the best performance on the GDI is obtained with BERT-base-cased.

To find the cut-off point for shortening the input, we apply a heuristic that relies on the analyses described in Section 4. Regarding model calibration, we found that, for all the fine-tuned models, ECE decreases considerably with calibration using temperature scaling (TS). For example, for the fine-tuned BERT-base-cased model without TS the ECE was 23.96, while with TS t= 2.28, ECE dropped to 6.3 in the GDI dataset. Similar experiments were done on ILI data with the IndicTransformer model set to fine-tune the T value. At t=1, we have an ECE of 20.09 for ILI while after calibrations at t=1.79, ECE dropped to 13.91.

In exploring input shortening criteria, we use the calibrated probabilities. We consider several shortening possibilities (the details are listed in Appendix A) and find that the best results are achieved with the same criterion in both data sets: probability drop. In other words, we stop the incremental classification once the model probability starts decreasing.

We add to this criterion the impact of the fragment length on the model accuracy, which is shown in Figure 1. The maximum accuracy for the GDI data is obtained at the length 4, while the peak is on length 7 for the ILI dataset. The trend is the same in both data sets, modulated by the length of the original utterances (longer in ILI).

The accuracy on shortened input shown in Table 2 is calculated on the first fragment that satisfies both criteria (model confidence and fragment length). Another finding that can be observed in Table 2 is that the gap between the full and the short input performance is smaller in models that perform better. This relationship applies only within a data set (not across languages).

6 Conclusion and Future Work

We have shown that dialect identification can be performed before the end of the given utterances. While we could not maintain the performance achieved with the full input, we have identified general criteria for making early guesses: language specific minimal length of the input (4 tokens for GDI, 7 for ILI) and language-independent change in the model confidence score (the first decrease in the confidence score).

In future work, we plan to continue improving the performance with early guessing by designing models specifically for this task. We plan to focus on unsupervised deep embedding clustering approaches (Xie et al., 2016; Goswami et al., 2020). We will also explore model calibration at training time and extend the analysis to other datasets (e.g. Arabic dialects).

[8] https://huggingface.co/models

Acknowledgements

References

Mohamed Ali. 2018. Character level convolutional neural network for german dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 172–177.

Gabriel Bernier-Colborne, Serge Léger, and Cyril Goutte. 2021. N-gram and neural models for uralic language identification: Nrc at vardial 2021. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 128–134.

Alina Maria Ciobanu, Shervin Malmasi, and Liviu P Dinu. 2018a. German dialect identification using classifier ensembles. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 288–294.

Alina Maria Ciobanu, Marcos Zampieri, Shervin Malmasi, Santanu Pal, and Liviu P Dinu. 2018b. Discriminating between indo-aryan languages using svm ensembles. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 178–184.

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.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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 the 28th International Conference on Computational Linguistics*, pages 1606–1617.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330.

Divyanshu Gupta, Gourav Dhakad, Jayprakash Gupta, and Anil Kumar Singh. 2018. Iit (bhu) system for indo-aryan language identification (ili) at vardial 2018. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 185–190.

Kushal Jain, Adwait Deshpande, Kumar Shridhar, Felix Laumann, and Ayushman Dash. 2020. Indictransformers: An analysis of transformer language models for indian languages. *arXiv preprint arXiv:2011.02323*.

Tommi Jauhiainen, Tharindu Ranasinghe, and Marcos Zampieri. 2021. Comparing approaches to Dravidian language identification. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 120–127, Kiyv, Ukraine. Association for Computational Linguistics.

Tommi Sakari Jauhiainen, Heidi Annika Jauhiainen, Bo Krister Johan Linden, et al. 2018. Heli-based experiments in swiss german dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. The Association for Computational Linguistics.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Avik Bhattacharyya, Mitesh M Khapra, Pratyush Kumar, et al. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *arXiv preprint arXiv:2005.00085*.

Nikola Ljubešić and Davor Lauc. 2021. Bertić-the transformer language model for bosnian, croatian, montenegrin and serbian. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 37–42.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in Neural Information Processing Systems*, 32:4694–4703.

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.

Tanja Samardzic, Yves Scherrer, and Elvira Glaser. 2016. Archimob-a corpus of spoken swiss german. In *Proceedings of the tenth international conference on language resources and evaluation (LREC 2016)*. European Language Resources Association (ELRA).

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.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 478–487.

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

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the vardial evaluation campaign 2017. In *Proceedings of the fourth workshop on NLP for similar languages, varieties and dialects*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, et al. 2018. Language identification and morphosyntactic tagging. the second vardial evaluation campaign.

Marcos Zampieri, Shervin Malmasi, Yves Scherrer, Tanja Samardžić, Francis Tyers, Miikka Silfverberg, Natalia Klyueva, Tung-Le Pan, Chu-Ren Huang, Radu Tudor Ionescu, et al. 2019. A report on the third vardial evaluation campaign. Association for Computational Linguistics.

A Explored Early Guessing Possibilities

In Tables 3 and 4, *'current'* is the current fragment under consideration. *prob()* is the calibrated probability. We compare the *prob(current)* with *prob(previous)* and *prob(next)*. As discussed, the fragment is the output of an incremental processing. The criteria checks will be done for each group of fragments that are associated with a particular sentence. Another input shortening criterion included is the labeling consistency. Here we check the consistency of predicted labels, *predicted label*. Each of these input shortening criteria is evaluated separately as well as in combination with each other. We consider the fragment that satisfies the input shortening criteria at the first position after a predefined length point, say, *m*. The value *m* will be different for each language and needs to be tuned based on performance metrics (accuracy/ F-score). The same input shortening criteria were evaluated for both GDI and ILI while considering different starting lengths *m*. For GDI we found optimal *m=4* while in ILI *m=7*. The results for each input shortening criterion are reported in Table 3 and Table 4. All the input shortening criteria are evaluated separately and some of the potential input shortening criteria are evaluated in combination.

Input Shortening Criteria	N	Accuracy
prob(current)>prob(previous):p1	4454	51.5% (2449)
prob(current)<prob(previous):p2	4130	47.49% (2257)
prob(current)<prob(next):p3	4048	44.9% (2134)
prob(current)>prob(next):p4	4605	55.2% (2624)
predicted label(current) equals predicted label(previous):l1	4549	52.5% (2496)
predicted label(current) equals predicted label(next):l2	4628	51.4% (2445)
p1 and l1	4143	50.35% (2393)
p2 and l2	3475	43.37% (2061)
p4 and l1	4354	53.57% (2546)
p1 and p4	4351	53.45% (2540)
p1 and p4	3024	37% (1762)

Table 3: Input Shortening Results with GDI. N= number of fragments that satisfy the criterion.

Input Shortening Criteria	N	Accuracy
prob(current)>prob(previous):p1	8096	71.24% (6435)
prob(current)<prob(previous):p2	7842	67.17% (6067)
prob(current)<prob(next):p3	7799	66.17% (5975)
prob(current)>prob(next):p4	8285	73.7% (6658)
predicted label(current) equals predicted label(previous):l1	7975	71.8% (6485)
predicted label(current) equals predicted label(next):l2	7964	72.44% (6543)
p1 and l1	7975	71.8% (6485)
p2 and l2	7240	66.44% (6001)
p4 and l1	8250	74.1% (6694)
p1 and p4	7946	67.3% (6076)
p1 and p4	7964	72.4% (6543)

Table 4: Input Shortening Results with ILI. N= number of fragments that satisfy the criterion.