Dialect Classification on LIA Norwegian Dataset using Deep Learning and Large Language Models

Anonymous ACL submission

Abstract

This paper will investigate the feasibility of identifying regional dialects in the Norwegian language, using language models. While languages are often considered as homogeneous entities, individual dialects can vary significantly in both vocabulary and syntax. Building on previous research conducted on Arabic dialect classification, this study aims to reproduce and expand on the results considering 17 Norwegian dialects. This research leveraged text samples collected in the LIA Treebank of Spoken Norwegian Dialects. Several Deep Learning models are employed to classify the dialects, including biLSTM, BERT, Roberta, and GPT 3.5. The Norwegian BERT model achieves the highest accuracy, outperforming other models in dialect classification. The findings highlight what can be achieved with carefully collected specialized data. Opportunities for future research lie in investigating alternative preprocessing techniques, model architectures, and the use of synthetic data.

1 Introduction

In the context of natural language processing, we often view each language, such as English, French, Arabic and Norwegian, as monoliths. However, upon closer inspection, this assumption is quite far from the truth. A single language in its spoken form can have many various regional accents. Sometimes, a language evolves differently across these regions, and each region develops a unique vocabulary and sometimes even syntactic particularities. When this happens, a region is said to have developed it's own dialect. We thought it would be extremely interesting to investigate whether a language model can identify which regional dialect a snippet of text is from, and so that is the research question we set out to solve. We have seen proof of this result being achieved with Arabic dialects by Talafha et al, and we have reproduced this result with 17 Norwegian dialects. We leveraged the LIA Treebank of Spoken Norwegian Dialects to conduct our research. Based on some related works described below, the models we chose to test during our experimentation were: biLSTM, ChatGPT 3.5, Nortram BERT, and Norwegian RoBERTa.

2 Related Work

From our research, it is evident that we are the first to apply traditional Deep Learning and Large Language model techniques to the task of Norwegian dialect classification on the LIA dataset. Therefore, due to this lack of research focus, we turned our attention to the more explored Arabic Dialect where we found 3 related papers to our task. The first was a paper written by Elaraby et Al, in which they explore the capabilities of Deep Learning models for Arabic dialect identification. In their paper, they empirically test 6 different deep learning methods on both binary and multi-way classification. Furthermore, Talafha et Al explore the use of pre-trained BERT models on country-level dialect identification. They do so by classifying 21,000 country-level labeled tweets covering all 21 Arab countries, achieving a maximum accuracy of 42.86% which is comparable to the results we have obtained on a 17 dialect classification task. Lastly, we were inspired by Abu Farha et Al benchmarking Transformer-based Language model paper where they explored different versions of pre-trained models such as BERT, RoBERTa, and ELECTRA to perform sentiment and sarcasm detection. This paper led us to discover different variations of Norwegian pre-trained models that would prove useful in our experiments.

While the cited research papers significantly advanced the field of dialect identification, their primary focus on Arabic dialects necessitated a careful adaptation of their experiments to suit the linguistic context of Norwegian.

3 Method

3.1 Dataset

3.1.1 The LIA Dataset

Our research builds upon the LIA Treebank of Spoken Norwegian Dialects, a project led by Øvrelid et al. at the University of Oslo. The dataset encompasses dialect recordings spanning 1950 to 1990, which were transcribed and annotated with part-of-speech (POS) tags. In addition, these annotations have undergone manual correction for accuracy and reliability. The dataset comprises 7536 speech segments and 77,701 tokens from the Nynorsk style of Norwegian encompassing 17 different dialects. Nevertheless, to ensure the overall quality of our dataset, we opted to exclude any speech segments with fewer than 3 words because these sentences were non-informative. Another important decision we made was to not exclude Norwegian stopwords from our analysis. This is because we believe the frequency in which different stopwords are used may be an indicator of the dialect they belong to. The figure below visually represents the class distribution of the 17 different dialects after pruning and is sorted based on their representation in the dataset.

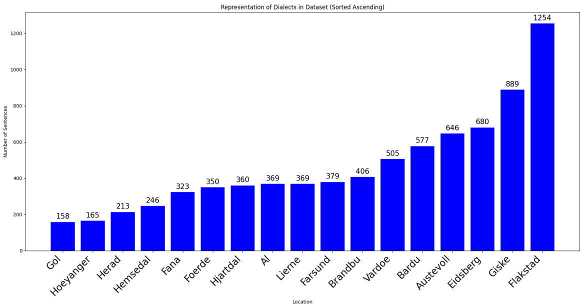


Figure 1: Representation of Dialects in the LIA Dataset after pruning.

Furthermore, we were interested in seeing how the 17 different dialects relate to one another in terms of shared vocabulary. A low rate of shared vocabularies would be indicative of many unique words in each dialect, whereas a high rate of shared vocabulary would indicate that these dialects may be more split along syntactic lines. To establish this crossover rate, we conducted an experiment consisting first of generating a vocabulary of all the words used in the sample sentences per dialect. Then for each pair of dialects, we calculated how many words were shared across vocabularies. As seen in Figure 5 in the Appendix, the overlap of non-identical dialects ranges quite evenly from 29%-52% overlap, with identical languages trivially scoring 100% overlap.

3.1.2 Experiment Setup

We were curious to see how different classifiers would perform as we gradually increased the number of dialects for classification in our experiments. This exploration served as the cornerstone of our research, where we systematically evaluated the performance of different classifiers across 3, 6, 12, and 17 distinct dialects.

To accomplish this, we first sorted the dialects based on the number of transcriptions. Following this sorting procedure, we then filtered the dataset to include the top k classes designated for classification exclusively. Our experiment is driven by two main goals. Firstly, we want to understand how changes in the amount of data affect the precision of the classifier. Additionally, we aim to challenge the models by having them learn from limited and potentially conflicting data. For our experiments, we randomly shuffle the filtered dataset and split it into 80 % training and 20% testing split, where 20% of the training set is used as a validation set during training time.

3.2 Models

3.2.1 biLSTM

Preparing Dataset for Training In the process of preparing the dataset for training, one common practice is to employ pre-trained word embeddings to represent words in the input sequences. However, for the specific context of the Norwegian language, utilizing existing word2vec models proved challenging due to the absence of certain dialectal words from our dataset. To address this issue, we attempted to train a custom word2vec model on our dataset, however, this strategy proved to be ill-suited, as unrelated words displayed unexpectedly high similarity scores meaning that the model was unable to capture the nuanced relationships between words in the LIA dataset.

As an alternative strategy, each word in the dataset was assigned and replaced with a unique integer identifier. This approach aimed to leverage the Keras Embedding layer in our model in order to learn the relationships between different words during training time. This same logic was applied to our experiment incorporating POS tags, where each tag received a unique integer, and the sentence representation involved concatenating the word-POS pairs as shown in X. Furthermore, since we are dealing with a multiclass classification task, each label is represented as a one hot encoded vector.

biLSTM Implementation Our biLSTM model was implemented using the Keras framework and takes advantage of the Sequential() functionality to build the model. To start, once each sentence is transformed it is padded up to the length of the maximum sentence length in our dataset to ensure uniform length. For the regular biLSTM model, each sequence has a padded length of 70, and for the POS concatenated model each sequence is padded to a length of 140. Furthermore, our Sequential() model is made up of an Embedding() layer, Bidirectional() layer, and a Dense() output layer. The Embedding layer takes in the padded sequence and outputs a dense vector of the same size. The Bidirectional LSTM layer is initialized with the same size as the output of the Embedding layer, and processes the embedded sequences in both the forward and backward direction. Finally, a Dense fully connected layer with a softmax activation function is used to produce the output probabilities for the given number of classes. Furthermore, the model is trained using categorical cross-entropy loss with an Adam optimizer and outputs an accuracy percentage. The model is trained for 100 epochs with a batch size of 128.

3.2.2 BERT and Roberta

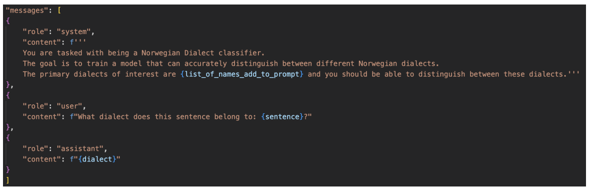
Preparing Dataset for Training For preprocessing, the data was formatted accordingly before one-hot encoding of the labels. After encoding, the pre-trained model's tokenizer, specifically designed for the Norwegian language was applied to the data. Using tokenization before training allows the model to process the text as words and convert the tokens into an integer representation. A sentence is broken down into a sequence of tokens while keeping hold of its semantic meaning. This aids the model in detecting context and relationships between words.

BERT and RoBERTa Implementation The models were taken from and trained using the Hugging Face platform, which provided a convenient and easy-to-use interface to work with pre-trained natural language processing models. Using these pre-trained Norwegian models is important as it allows the use of existing Norwegian knowledge. These pre-trained models have learned Norwegian language patterns and contextual information from a large amount of unannotated raw data. In fact, the Norwegian BERT model was trained on the Norwegian Colossal Corpus comprising 48.9 GB of Norwegian language data, and the Norwegian RoBERTa model was trained on the Oscar database comprising 4.7 GB of Norwegian language data. Furthermore, we utilize transfer learning to create our classification models as we extend the functionality of the pre-trained models. In order to do this, we first load the model using the `AutoModelForSequenceClassification.from_pretrained()` function and specify the number of classes that our model should distinguish from. Following this, we instantiate our training arguments and define a function named `compute_metrics` to compute the metrics at the end of each evaluation epoch. The `compute_metrics` function takes the model predictions, the `logits`, and true labels during evaluation, converts them to class indices, and computes the accuracy. Furthermore, the model was then trained using a trainer for hyper-parameter tuning, using the Trainer() function, to increase the model's accuracy. The BERT classifier was trained four times to assess its effectiveness in differentiating an increasing number of dialects.

These steps were then repeated once more to create the Norwegian pre-trained RoBERTa-base classifier utilizing another pre-trained Norwegian model "`patrickvonplaten/norwegian-roberta-base`". The steps taken in constructing this model were identical to the BERT classifier, however, utilized the respective RoBERTa tokenizer and model.

3.2.3 GPT3.5

Preparing Dataset for Training The preparation for the GPT3.5 dataset is different from the two other models as this model takes a prompt format to train on. The prompt format consists of three components: system prompt, user prompt, and assistant prompt. The system prompt sets context and guidelines, the user prompt directs queries, and the assistant prompt instructs the model on response construction. This trio ensures a focused and structured interaction, guiding the generation of contextually relevant responses. We used the following prompt to construct our training and validation data, where "`list_of_names_add_to_prompt`" represents the dialects to be classified, "`sentence`" represents the current sentence we are classifying, and "`dialect`" represents the truth value.



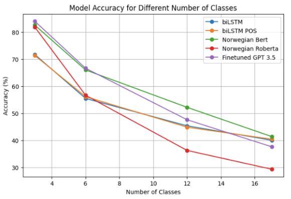
```
"messages": [
{
    "role": "system",
    "content": f'''
    You are tasked with being a Norwegian Dialect classifier.
    The goal is to train a model that can accurately distinguish between different Norwegian dialects.
    The primary dialects of interest are {list_of_names_add_to_prompt} and you should be able to distinguish between these dialects.'''
},
{
    "role": "user",
    "content": f"What dialect does this sentence belong to: {sentence}?"
},
{
    "role": "assistant",
    "content": f"{dialect}"
}
]
```

Figure 2: Prompt Strucutre for GPT Fine Tuning

GPT3.5 Implementation The GPT-3.5 implementation details and model weights are not publicly accessible. Consequently, we leveraged OpenAI's API to execute four fine-tuning tasks for each classification task. This was achieved through the use of the "`client.fine_tuning.jobs.create()`" function, wherein we provided our transformed training and validation prompt dataset. We set the number of epochs to the maximum value, 3, and utilized OpenAI's functionality to automatically determine an appropriate learning rate.

4 Results

4.1 Accuracy Results



Models	3 Label (%)	6 Label (%)	12 Label (%)	17 Label (%)
biLSTM	71.68	55.28	45.37	40.05
biLSTM + POS	71.33	56.14	44.86	40.43
BERT	82.53	66.00	**52.19**	**41.38**
Roberta	81.81	56.66	36.28	29.35
GPT3.5	**83.97**	**66.61**	47.61	37.62

Figure 3: Model Accuracies for Different Number of Classes

Figure 3 shows the results achieved by all the models on the dataset for each dialect classification experiment. As can be seen in Figure X, most models achieve good results on the 3 Dialect Classification task. However, as each task becomes more challenging a downward trend is observed for each model. Overall, on the lesser complex tasks the GPT3.5 and Norwegian BERT models stand out as they have an 11% accuracy difference compared to the other models.

However, as the complexity of the tasks grows by adding more dialects, the Norwegian BERT model performs better than any other model on this dataset. Surprisingly, the performance of the Norwegian RoBERTa model was inferior to our expectations. Despite being trained on a substantial 4.7 GB of Norwegian data, it failed to adequately capture the intricacies of the dialects. This observation reinforces the argument that for Language Model Models to be effective, they must undergo training on meticulously curated data.

Furthermore, regarding the performance of the BiLSTM, it seems that the addition of the POS tags to the embeddings did not have as much of an effect as we were hoping for as there was not a significant gain in accuracy from its addition. More importantly, the biLSTM models outperformed the majority of the LLMs on the most complex task from our experiments.

Moreover, our analysis revealed a noteworthy pattern as the complexity of the task increased. In fact, it seems that the top-performing models converge towards a shared performance trajectory. This is shown by the low variation of performance on the 17 dialect classification task.

4.2 Computational Cost

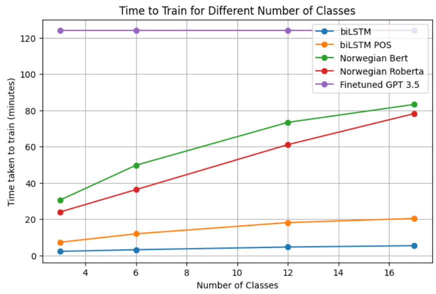


Figure 4: Time to Train separated by Model

The advent of transformer-based language models marked a significant leap forward in Natural Language Processing, driving a surge in innovation and delivering state-of-the-art results across various tasks. Nevertheless, the transformative efficacy of these models highlights a critical concern: the substantial computational costs inherent in their development. As we strive for improved performance, questions naturally arise about how accessible these models are and the substantial time investment needed for their training and fine-tuning. This leads us to carefully explore whether the benefits they bring outweigh practical considerations, especially when compared to more traditional deep learning models in tasks like classification.

Figure 4 shows the time needed to fine-tune each of the models across the 4 different classification tasks. The experiment results highlight a crucial relationship between the complexity of language models and the time it takes to train them. Notably, simpler models like biLSTM outpace their more sophisticated counterparts, with GPT-3.5 standing out as the slowest in training. Furthermore, a clear pattern is revealed when analyzing the data, an increase in model complexity corresponds to a prolonged training duration. This is evidenced by a significant time gap between models such as GPT3.5 and BERT. Moreover, the figures visually confirm this relationship, portraying a consistent uptick in training time as tasks become more complex.

In resource-constrained scenarios, particularly for challenging tasks such as 17-dialect classification, opting for biLSTM models proves pragmatic. Their faster training pace, coupled with performance comparable to larger language models, positions biLSTM as an optimal choice when faced with efficiency challenges and limited resources.

5 Discussion and Conclusion

In this paper, we demonstrate that the efficacy of dialect classification is contingent upon the task's complexity, specifically in terms of the number of potential classes. For instance, the GPT3.5 model yields high accuracy on the 3-dialect classification task, making it plausible to assert that a dialect can be identified a majority of the time. However, as the task complexity increases, the models' accuracies experience a rapid deterioration. Overall, our findings highlight the Norwegian BERT model as the top-performing and most consistent among the models we tested. However, it is important to note that the biLSTM models performed almost as well as the complexity of the classification tasks grew.

Two key limitations characterize this study: dataset size and data quality. Given the relatively small size of the LIA dataset, it is reasonable to infer that with a larger dataset, the models might have exhibited improved performance. Moreover, as non-Norwegian speakers, we assume that the transcriptions adequately capture the characterstics of each dataset.

To extend this work, future research could focus on expanding the dataset both in terms of size and diversity, encompassing a broader range of dialectal variations. Additionally, investigating the impact of different pre-processing techniques and model architectures on dialect classification may offer insights into optimizing model performance. Addressing these aspects would contribute to refining the accuracy and applicability of dialect classification models in real-world scenarios.

6 Statement of Work

Liamo: I conducted an analysis of the dataset, wrote the biLSTM models and GPT model experiments and I contributed to writing the report.

Justin: I set up the dataset for ease of use, wrote the BERT and RoBERTa model experiments and I contributed to writing the report.

Abe: I conducted a large-scale exploration of publicly available datasets which included text-based samples of a language labeled by regional dialect. I contributed to deciding the structure of the experimentation, and I also conducted the analysis comparing the dialect vocabularies to each other. Additionally, I contributed to writing the report.

7 Appendix

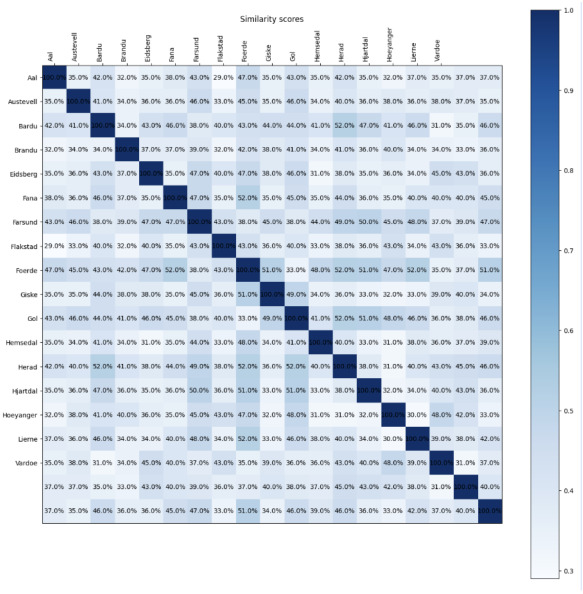


Figure 5: Similarity Scores of DIfferent Dialects

Github Link to our project: https://github.com/L-Pen/Norwegian-Dialect-Classification.git

References

Ibrahim Abu Farha and Walid Magdy. 2021. Benchmarking transformer-based language models for Arabic sentiment and sarcasm detection. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 21–31, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Abdullah Aziz Sharfuddin, Md. Nafis Tihami, and Md. Saiful Islam. 2018. A deep recurrent neural network with bilstm model for sentiment classification. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4.

Mohamed Elaraby and Muhammad Abdul-Mageed. 2018. Deep models for Arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 263–274, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Hugging Face. n.d. Fine-tune a pretrained model. https://huggingface.co/docs/transformers/training.

Nasjonalbiblioteket AI Lab. 2021. NbAiLab/Notram-Bert-Norwegian-cased-080321 · Hugging Face. https://huggingface.co/NbAiLab/notram-bert-norwegian-cased-080321.

Lilja Øvrelid, Andre Kåsen, Kristin Hagen, Anders Nøklestad, Per Erik Solberg, and Janne Bondi Johannessen. 2018. The LIA treebank of spoken Norwegian dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Python Package Index. 2023. Conllu. https://pypi.org/project/conllu/?fbclid=IwAR1AECdteVvkwh7DRInPv_HdNX-c324M5BQd7nNPhvvAY3ydoJyznBAoFBY.

Bashar Talafha, Mohammad Ali, Muhy Eddin Za'ter, Haitham Seelawi, Ibraheem Tuffaha, Mostafa Samir, Wael Farhan, and Hussein T. Al-Natsheh. 2020. Multi-dialect arabic bert for country-level dialect identification.

Christian Versloot. 2023. Bidirectional lstms with tensorflow and keras. GitHub repository.

Patrick von Platen. 2021. Patrickvonplaten/Norwegian-Roberta-base · Hugging Face. https://huggingface.co/patrickvonplaten/norwegian-roberta-base.