The Impact of Training Data Size on Classifier Performance for Swiss German Dialect recognition

Cedric Osei-Akoto
Zurich University of Applied Sciences,
Winterthur
oseiaced@students.zhaw.ch

Sithursan Sivasubramaniam
Zurich University of Applied Sciences,
Winterthur
sivassit@students.zhaw.ch

Abstract

The optimisation of dialect detection for swiss german dialects is a complex task. In this context we audited the performance impact of different training data set sizes for the detection of the bernese and eastern swiss dialects. The examination was conducted with four experiments, in which the training data set sizes deferred by factor 2. Throughout the experiments the performance of the classifier was measured with the f1-score, which consistently indicated low impact on the amount of correct predictions.

1 Introduction

Dialect detection is considered an important research area within the speech and natural language processing realm. The main objective of dialect detection is to classify a language dialect based on an audio file and with as little meta data as possible. Due to this goal classifiers serve as the most common tool to solve the task of dialect detection. When designed properly such a classifier has the ability to enhance solutions for additional applications. Speech recognition, language translation and sentiment analysis in audio files are prime examples of tasks, which can profit from a good dialect classifier. But how do this advancements in NLP research perform on our own dialects? In order to better grasp this question we explicitly had a closer look at the swiss German dialect. Generally it is considered a collection of distinct Alemmanic dialects, which varies considerably by region spoken in. In order tackle the question of how good dialect detection works on swiss german data, we specifically reintroduced a basic pipeline to mimic today's technology.

1.1 Research goal and scope

Since the initial question is defined rather broad we focused on creating a classifier, which can differentiate forms of Swiss German dialects. Specifically this paper audits the hypothesis if more training data for such a classifier translates directly into an overall better f1 score. In order to thoroughly inspect this hypothesis we also streamlined the classification task to only differentiate between two Swiss German dialects. Based on large geographic and linguistic differences we choose the Bernese and eastern swiss dialects for our experiments.

1.2 Research goal metric

Due to time constraints and limited translated text data of the project, we decided to focus on the f1 score of our classifier as evaluation metric instead of the conventional Word error rate. The minimal requirement we aimed to reach in our experiments was an f1 score over 0.5. Such an f1 score would indicate that our classifier is capable of learning the distinct difference between the two dialects.

2 Data

The Data used for training and testing of our classifier was provided by the Schweizer Dialektsammlung (SDS-200). It contains Swiss German Audio files and a reference table bearing the corresponding german translation, speakers age, gender, dialect, canton, region and zipcode of the dialects origin.

2.1 Data preparation

In order to effectively access our hypothesis we conducted 4 distinct test with different train data set sizes. The Table 1 in the appendix gives a rough idea about the distribution of values.

2.1.1 Process behind data preparation

Initially we filtered the given train data with the reference data table by bernse and eastern swiss dialect to extract the essential data for our experiments. Followed by the extraction was a random pick of a number entries coherent to the different experiment train data set size numbers. In the next step we transformed the corresponding .flac and .mp3 audio files from the test and train data set to numeric vectors/matrices using the Wave2Vec module provided on huggingface by facebook. This last step enabled us to later compute the data entries as input in our Convolutional neural network.

3 System Design

Based on the description of our research goal, our main focus was centered purely on the effects of the different training data set sizes on the f1 score. Hence our experiment pipeline was designed around this very concept. Consequently our experiment Pipeline is constructed of a preprocessing, Wave2Vec and convolutional neural network blocks as can be seen in the visualization below.(Figure 1) The pipeline makes use of structured preprocessing and Wave2Vec components, by relocating copies of the original and preprocessed audio files. Both preprocessing and Wave2Vec elements have been optimized for fast computation performance with Google colab's GPU's and libraries such as Swifter, enabling us to rerun the same experiments at a fast rate.

3.1 Convolutional Neural Networks

For the classifier a rather simplified approach was chosen due to the circumstances. The visual below shows the underling build of our CNN. (Figure 2)

Our convolutional neural network is constructed with two convolutional layers, max pooling layers and two fully connected layers, which all extract hierarchical features from the input data. The convolutional layers respectively produce 16 and 32 output channels and both have kernels of size 3. The following pooling layers are then used to reduce the spatial dimensionality of the system while maintaining essential features of the inserted data. The additional two fully connected layers are responsible for transforming high-level features into class scores. This is done with help of the 64 output features of the first layers and the utilization of the ReLU activation function, which introduces the networks non-linearity. At that point of the system a dropout layer and linear layer is introduced to mitigate the risk of overfitting of the classifier.

3.2 Monitoring

In order to have an exact overview of the experiment progress and results, the entire system is monitored over the weights and biases framework.

4 Results

As mentioned above 4 distinct experiments were conducted during this research project. Each of this experiments was executed 3 times with different random seeds to ensure the consistency of our results. The visualization below (Figure 3) summarizes the monitored results over all experiments.

As can bee seen in the line graph (Figure 4), the created models had a significantly lower f1 score than the initially aimed 50%. In addition we measured an accuracy fluctuating between 42% and 55%. This implies that True Positive and True Negative values were determined with a rate of about 50%, while False Negatives and False Positives were merely around 18%.

5 Conclusion

Based on both accuracy and f1 score computed over all four experiments, we can conclude that our hypothesis has to be rejected. The rather low scores of the classifier over all data set sizes imply that the classifier was not able to learn the dialect detection task properly. Due to the accuracy of around 50% it can be anticipated that the model only was able to learn one of the two dialects.

As consequence further studies are necessary to enhance the understanding of the gained results of this method. Current research findings imply that Wave2Vec may not be the optimal selection for the dialect detection task as an possible explanation. Hence implicitly suggesting that Wave2Vec's architecture may not be optimized for dialect classification, due to it lack of capturing subtle differences in pronunciation and intonation between different dialects. In this context further research is needed to explore alternative approaches that can better capture these nuances. A further important fact which has to be noted is that our CNN model was not modified for the specific data set. By modifying the hyperparameters or adding a Long Short-Term Memory (LSTM) layer to the model at least the baseline results our model could be significantly be improved.

6 Authorship and Contribution

This paper was written by Cedric and Sithursan, with both authors contributing equally to the writing and revision of the text.

7 Appendix

7.1 Tables

Experiment	Bernese dialect	Eastern swiss dialect	Train	Validation	Testing
Experiment 1	200	200	320	80	7030
Experiment 2	400	400	720	160	7030
Experiment 3	800	800	1280	320	7030
Experiment 4	1600	1600	2560	640	7030

Table 1: Experiment data set sizes

7.2 Visualizations

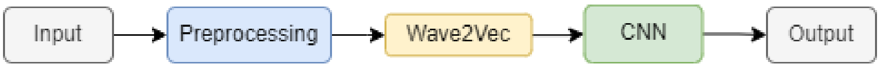

Figure 1: Design of the experiment pipeline

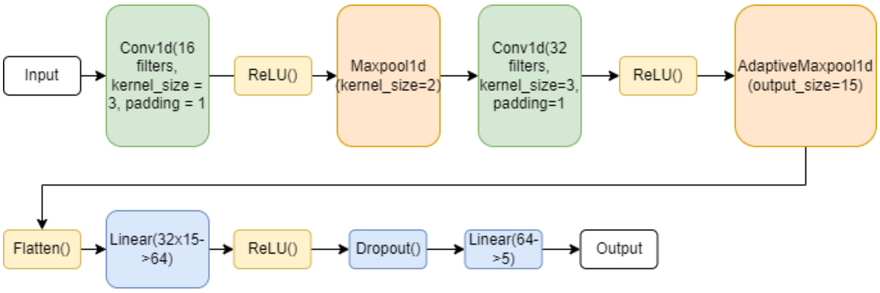
Input
Conv1d(16 filters, kernel_size = 3, padding = 1
ReLU()
Maxpool1d (kernel_size=2)
Conv1d(32 filters, kernel_size=3, padding=1
ReLU()
AdaptiveMaxpool1d (output_size=15)
Flatten()
Linear(32x15->64)
ReLU()
Dropout()
Linear(64->5)
Output

Figure 2: Design of the 2 layer CNN model

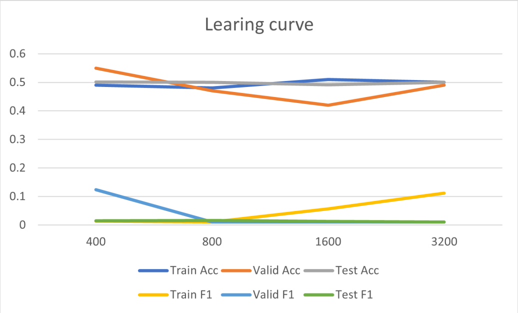

Figure 3: Learning curve with all 4 Experiments

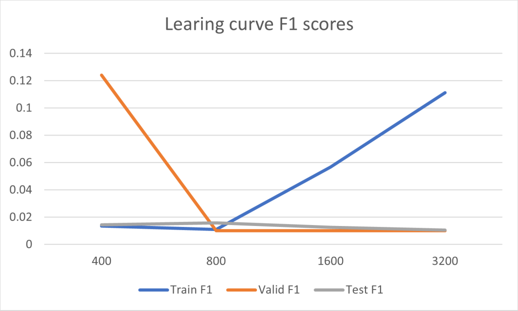

Figure 4: Learning curve with the F1 scores

7.3 References

References

[1] S. Stucki and P. Randjelovic, *Exploring Wav2Vec2 Pre-Training on Swiss German Dialects using Speech Translation and Classification*. ZHAW, 2021.

[2] M. Pluss, M. Hurlimann, M. Cuny, A. Stockli, N. Kapotis, J. Hartmann, M. Ulasik, C. Scheller, Y. Schraner, A. Jain, J. Deriu, M. Cieliebak, and M. Vogel, *SDS-200: A Swiss German Speech to Standard German Text Corpus*. arXiv preprint arXiv:2205.09501, 2022.