COMP9444 Project Summary

Spoken Language Identification

Group - 'To Be Decided'
Members - Joel Bryla (z5361331), Andrea Dobles (z5394074), Mikkel Endresen (z5304263), Gajendra Jayasekera (z5260252), Gayathrie Vijayalingam (z5193713)

I. Introduction

Automatic language identification (LID) is a challenging problem in speech signalling processing. The task is to identify the language of a given audio clip. This is an important first step in many applications, such as speech recognition, multilingual voice assistants, transcription and translation services, and content filtering on social media; but may also serve an important purpose on its own, such as in emergency services, where transferring the call to an operator who can speak the caller's first language may save lives (Das & Roy, 2019). In this project, we aim to improve upon existing deep learning models and combine features of these models to create a model that is both fast and accurate.

II. Literature Review

Older LID systems use parallel phone recognition followed by language modelling (PPRLM), which classifies languages based on distinct speech sounds (phones), or i-vectors, which are compact representations of a whole utterance using a Gaussian Mixture Model (GMM) supervisor. Although accurate, PPRLM systems need more processing time and labelled data while i-vectors work better with longer utterances and have some latency as a majority of computation occurs at the end of the whole utterance (Cordoba et al., 2003, Lopez-Moreno et al., 2016).

The current state-of-the-art approaches to LID use deep learning. There are two main systems: end-to-end systems where feature extraction and classification are done in one system, and hybrid i-vector or PPRLM and NN systems that use more resources but are more accurate. This tradeoff is important to take into consideration for real-time applications, where we may need to favour speed over accuracy. For this reason, many recent systems are end-to-end NN systems (Das & Roy, 2019).

During the feature extraction stage, different linguistic cues are extracted from the audio and used during the classification stage. Acoustic phonetic features can be extracted by several methods, but the most widely-used one is mel-frequency cepstral coefficients (MFCCs), as the nonlinear mel scale that it uses better approximates the human ear scale (Das & Roy, 2019).

Several types of neural networks have been implemented to solve LID problems, including feed-forward neural networks (Lopez-Moreno et al., 2016 Richardson et al., 2015), convolutional neural networks (CNN) (Montavon, 2009, Lei et al., 2014), and recurrent neural networks (RNN) (Gonzalez-Dominguez, et al., 2014). Some systems, like the convolutional recurrent neural network (CRNN) proposed by Bartz et al. (2017), convert audio to spectrograms and work better on noisy data.

III. Models and Methods

Our initial research revolved around the exploration of the ShallowCNNTDNN model, as detailed in the paper, Deep Learning for Spoken Language Identification (Montavon, 2009). This model incorporates the Time-Delay Neural Network (TDNN) architecture, which is renowned for its proficiency in sequence modelling tasks. This architecture finds widespread application in domains such as speech processing and natural language processing (NLP), where the handling of temporal patterns and sequential data is important. Along with a convolution neural network (CNN) that extracts features especially in images (our spectrograms) and the use of TDNN was great motivation to investigate this model. The Deep-CNNTDNN, which was detailed in the same paper, is an extension of the shallow model with more convolutional layers. The addition of more layers improved the accuracy of the model.

Our third model, to improve upon the Deep-CNNTDNN, was a CRNN model. It utilised a convolutional net as the feature extractor and a BLSTM to combine the features over time. It had seven convolutional layers, five of them followed by maxpool, all using batch normalisation, and then two LSTMs as the BLSTM before a fully connected layer to the output. It is based on a research article that showed significant improvement when adding a RNN to CNN (Bartz et al., 2017). We used the Adam optimizer with a learning rate scheduler that started at 0.0001 and multiplied the learning rate by 0.1 for every five epochs. The initial learning rate was chosen after hyperparameter tuning. We decided to implement the model with a learning rate scheduler because of the volatility in the validation error rate.

Testing revealed the Deep-CNNTDNN took 1hr to train and attained an accuracy of 79% whilst the CNN-BiLSTM took 4hrs and attained an accuracy of 93% (explored in depth in Results). The difference between the training time, accuracy and subsequently the file size (77KB vs 100MB) prompted the experimentation and development of a 4th model, the CNN-LSTM. This model seeks to find a satisfactory compromise between speed and accuracy to enable a scalable model capable of being trained on more languages. The architecture takes inspiration from both Deep-CNN and BiLSTM finding a midpoint of two models.

IV. Experimental Setup

All data used in the project is from the Voxlingua107 dataset (at https://bark.phon.ioc.ee/voxlingua107/) which is specifically designed for spoken language identification. It contains over 6000 hours of audio between 107 languages and uses an automatic collection and labelling system, first finding key words for

a language from language specific wikipedia pages then querying youtube with these terms. The videos are run through a speech detection system to extract segments with speech and then to remove mislabelled samples a language embeddings classifier is trained and samples which appear as outliers are removed. The authors claim this results in a 98% correctly classified dataset.
We decided to train our models on 5 languages: English, Danish, Swedish, Arabic and Mandarin Chinese. Differentiating between Danish and Swedish was expected to be the larger challenge due to their similarities. Including Arabic and Mandarin would allow us to evaluate the model on the expected easier task of differentiating between vastly different languages.

The data for the 5 languages was retrieved and cut into 5 second clips which were converted to Mel Spectrograms. The samples were then balanced such that there was an equal number of each language. This resulted in 70000 samples, about 100 hours. Which was split 80:20 between the training set and validation set. With 57092 samples and 14273 samples respectively. The dataset also contained a separate smaller development set which was verified by two humans to be the labelled language. This data was transformed and balanced in the same way as the previous sets and used as a final test set, containing 165 samples, ~0.25 hours.

For all models negative log loss was used with the optimiser. For human evaluation an accuracy metric on each set was given for context when evaluating training.

V. Results

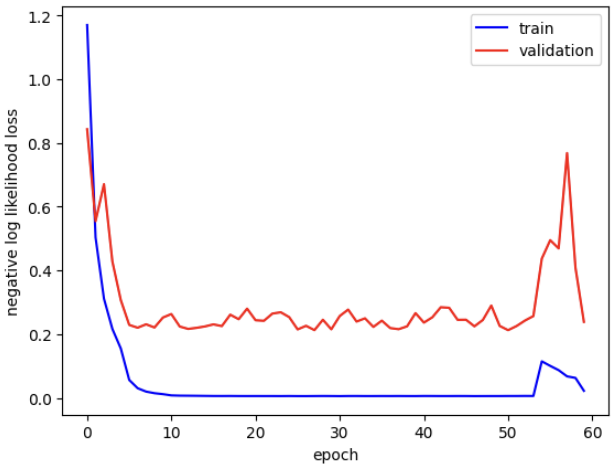
The performance of the ShallowCNNTDNN varied with multiple parameters. Key results showed better performance with Adam (Adaptive Moment Estimation) as the optimizer over SGD (Stochastic Gradient Descent) which the paper's model was based on. This was the basis on why Adam was chosen for the other models as well. The model was first trained and tested for three languages (English, Arabic, and Chinese Mandarin). After varying the parameters the best performance for the Shallow model was 70% in three languages and 57% for five languages (first three, Danish and Sweden) . On the other hand, the best Deep CNNTDNN had a validation accuracy of 79% and a test accuracy of around 73% on five languages. It is a bit lower compared to the accuracy that Montavon et al. (2017) is able to get on a different dataset (83.5%), but considering they used three languages, the loss of accuracy with increasing to five languages is expected.

The CNN-BLSTM had a validation accuracy of 92.7% and a test accuracy of 88.4%. Its training error rate converged while the validation rate hovered at around 0.2. In addition, the confusion matrix showed that the most errors came by predicting north-germanic and germanic languages. It has over twenty six million trainable parameters that took over four hours to train with 60 epochs. The final model can be seen below in Figure 1.

The CNN-LSTM returned an accuracy of 82.7% on the 5 trained languages with a learning rate of 0.0001 and a weight decay of 0.01 over 50 epochs struggling to converge the error. Given more time, the

implementation of a scheduler with varying learning rates may have helped improve the volatility by escaping the local minima. Closer analysis revealed the model struggled to distinguish between Danish and Swedish however shows proof of concept as it reduced the training time by 50% to 2hrs and the file size by 62% to 38MB.

Model	CNNBiLSTM
Learning Rate	0.0001
Weight Decay	0.001
Optimizer	adam
LR_Scheduler	StepLR_step=5_gamma=0.1
Dropout	nan
Batch Size	64
Training Epochs	60
Train Loss	0.022
Validation Loss	0.238
Test Loss	0.454
Validation Accuracy	92.777
Test Accuracy	88.485



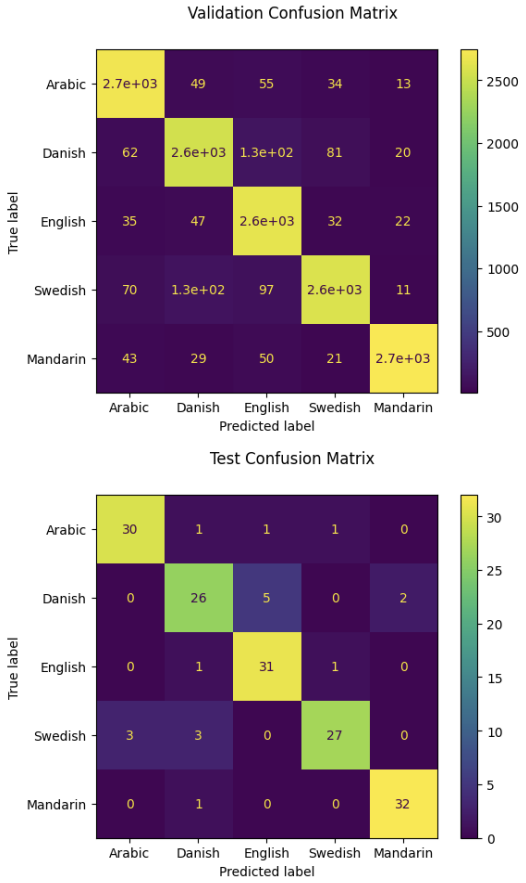


Figure 1. Final CNN-BLSTM Model

VI. Conclusions

We created a demo to test the Deep CNNTDNN and the CNN-BLSTM. Based on the demo, our model performs reasonably well at distinguishing the five languages in a real-world application. To improve our model, more parameters aside from learning rate and weight decay could be tested, as well as different architectures. An increased set of samples for each language class would also help the model better distinguish edge cases such as accents. More preprocessing of the audio samples, such as voice activity detection and noise removal can improve learning as well, allowing the model to make good predictions even when there is background noise or music. Finally, our models were only trained on five languages, so increasing the number of languages that the model is trained on would increase its usability for real-world applications.

VII. References

Bartz, C., Herold, T., Yang, H., & Meinel, C. (2017). Language identification using deep convolutional recurrent neural networks. In Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14–18, 2017, Proceedings, Part VI 24 (pp. 880-889). Springer International Publishing.

Cordoba, R., Prime, G., Macias-Guarasa, J., Montero, J. M., Ferreiros, J., & Pardo, J. M. (2003). PPRLM optimization for language identification in air traffic control tasks. 8th European Conference on Speech Communication and Technology (Eurospeech 2003). https://doi.org/10.21437/eurospeech.2003-732

Das, H. S., & Roy, P. (2019). A deep dive into deep learning techniques for solving spoken language identification problems. Intelligent Speech Signal Processing, 81–100. https://doi.org/10.1016/b978-0-12-818130-0.00005-2

Gonzalez-Dominguez, J., Lopez-Moreno, I., & Sak, H. (2014). Automatic language identification using long short-term memory recurrent neural networks.

Lei, Y., Ferrer, L., Lawson, A., McLaren, M., & Scheffer, N. (2014). Application of Convolutional Neural Networks to Language Identification in Noisy Conditions. In Odyssey.

Lopez-Moreno, I., Gonzalez-Dominguez, J., Martinez, D., Plchot, O., Gonzalez-Rodriguez, J., & Moreno, P. J. (2016). On the use of deep feedforward neural networks for automatic language identification. Computer Speech & Language, 40, 46–59. https://doi.org/10.1016/j.csl.2016.03.001

Montavon, G. (2009). Deep learning for spoken language identification. In NIPS Workshop on deep learning for speech recognition and related applications (pp. 1-4).

Richardson, F., Reynolds, D., & Dehak, N. (2015). Deep neural network approaches to speaker and language recognition. IEEE signal processing letters, 22(10), 1671-1675.

Valk, J., & Alumäe, T. (2021, January). VoxLingua107: a dataset for spoken language recognition. In 2021 IEEE Spoken Language Technology Workshop (SLT) (pp. 652-658). IEEE. https://arxiv.org/abs/2011.12998