Laboratory Assignment 2 - Native Language Identification challenge

Group 13
Diogo Miranda - 96190 - diogomiranda26@tecnico.ulisboa.pt
Guilherme Oliveira - 96221 - guilherme.a.oliveira@tecnico.ulisboa.pt

***Abstract*—In this laboratory assignment, we built different systems for native language identification using a set of target native L1 languages: Chinese, German, Hindi and Italian. The ultimate goal is to develop the model most suitable for this task. The structure of this report is split between classical and modern approaches.**

I. Classical models based on conventional features

We implemented MFCC feature extraction with several additional components: Delta and Double-Delta computation, Shifted Delta Cepstrum (SDC), Voice Activity Detection (VAD), and Cepstral Mean and Variance Normalization (CMVN). Two distinct models were created:

1) The first model, `mfcc39_d_dd_vad_cmvn`, utilized Delta and Double-Delta features, followed by VAD and CMVN.
2) The second model, `mfcc_sdc_vad_cmvn`, employed SDC features, followed by VAD and CMVN.

As a preprocessing step, mean removal was applied to both models.

The model in this baseline is extremely simple: we train an individual Gaussian Mixture Model (GMM) for each L1 language using the extracted features. During prediction, for a given test audio sample, we compute the log-likelihood for each GMM model. The L1 native language is identified as the one corresponding to the GMM model that provides the highest likelihood.

For each model, we experimented with different values of `n_gauss` and `init_params`. We tracked the accuracy and the time taken by each model, and summarized the results in a table. All tests were done using the train100 dataset.

Model Configuration	`n_gauss`	`init_params`	Accuracy (%)	Time (s)
`mfcc39_d_dd_vad_cmvn`	64	kmeans	61.93	496
`mfcc39_d_dd_vad_cmvn`	64	k-means++	66.48	70
`mfcc39_d_dd_vad_cmvn`	128	kmeans	65.34	852
`mfcc39_d_dd_vad_cmvn`	128	k-means++	65.91	568
`mfcc_sdc_vad_cmvn`	64	kmeans	72.73	580
`mfcc_sdc_vad_cmvn`	64	k-means++	72.16	316
`mfcc_sdc_vad_cmvn`	128	kmeans	76.14	1092
`mfcc_sdc_vad_cmvn`	128	k-means++	69.32	556

TABLE I
Models performance

As shown in Table I, using a higher `n_gauss` significantly increases the time taken to run the model, but it also improves the accuracy. Utilizing k-means++ greatly reduces the time taken and, for the first model, results in a slight increase in accuracy. However, for the second model, k-means++ results in lower accuracy.

The best model was `mfcc_sdc_vad_cmvn` using 128 `n_gauss` and `kmeans` as `init_params`. Because this was the best model, we performed additional tests, mainly using different types of VAD. The results are presented in the table below.

VAD Type	Accuracy (%)	Time (s)
MFCC0-Based(default)	76.14	1092
GMM-Based	72.16	1076
Energy-Based	75.57	1048

TABLE II
Effect of VAD on Model Performance

The Voice Activity Detection (VAD) methods employed in our experiments utilize different techniques to segment speech from non-speech segments within the audio data. The Energy-Based method determines speech segments based on energy levels, while the GMM-Based method utilizes Gaussian Mixture Models to classify segments. The MFCC0-Based method focuses on the first Mel-frequency cepstral coefficient (MFCC0) to detect speech segments.

As indicated in Table II, the best model identified was `mfcc_sdc_vad_cmvn` using MFCC0-Based VAD. Given its superior performance, we proceeded to train this model using the entire dataset and submitted it to Kaggle. This model achieved an accuracy score of 76% on the evaluation dataset.

II. Modern systems using pre-trained models

A. *Using pre-trained speaker embeddings (x-vectors)*

We leveraged x-vectors as features and trained an SVM model for classifying target languages. Specifically, we incorporated embeddings from two pre-trained models: *spkrec-ecapa-voxceleb* and *lang-id-voxlingua107-ecapa*. We achieved an accuracy of 84.7% and 92.6%, respectively, on the development dataset. These outcomes were anticipated due to the specific training objectives of each x-vector model. While *spkrec-ecapa-voxceleb* was designed for speaker verification tasks, *lang-id-voxlingua107-ecapa* was tailored for language identification across a diverse corpus of 107 languages.

Although several other x-vector models were accessible via the Hugging Face Platform, many were unsuitable for language identification or lacked the extensive language coverage of *lang-id-voxlingua107-ecapa*. As a consequence, we didn't pursue any more experiments and selected the model using this x-vector as the most suitable for our native language identification task.

B. Using self-supervised pre-trained models

Next, we evaluated pre-trained upstream models with different configurations to compare their performance on our downstream task. Table III contains the results obtained for each model using the same initial configuration: train100 dataset, 1000 total steps and Mean Pooling.

Model Configuration	Performance on the development set (%)
fbank	25.00
hubert	82.39
xlsr_53	85.80
wavlm	91.47

TABLE III
MODELS PERFORMANCE

The time taken to train the models wasn't taken into account, as the models were trained using different setups, including Google Colab and personal computers. Due to the computational expense of training the models, we selected the best-performing model and conducted further experiments only on it. The best model identified was *wavlm*. Subsequently, we explored the larger model variant, *wavlm_large*, and experimented with different pooling methods, including MeanPooling, MaxPooling, and AttentativePooling. These tests were conducted using the *train100* dataset and 1000 iterations. The results are summarized in the table below.

Pooling Method	Performance on the development set (%)
Mean Pooling	97.16
Max Pooling	97.16
Attentative Pooling	97.72

TABLE IV
MODELS PERFORMANCE

As evidenced by the table, the optimal model utilized Attentative Pooling, achieving an accuracy of 97.72%. This equates to merely four incorrect predictions on the development dataset.

Finally, having selected the best method, we opted to perform a final training using the entire training dataset for 5000 iterations. This effort resulted in an accuracy of 98.86%, corresponding to only two incorrect predictions, although it took considerably longer to train the model. Upon inspection of the log.log file, we observed that the training accuracy continued to increase until the final iteration. This suggests that further training iterations could potentially lead to improved results. Submitting this model on Kaggle gave us a evl accuracy of 98%.

III. CONCLUSION

After experimenting with both traditional and modern approaches, we found that the best modern models consistently outperform their traditional counterparts. This underscores the effectiveness of deep learning models in automatically learning features compared to manually curated traditional features. Specifically, for our Native Language Identification task, the self-supervised pre-trained wavlm_large model proved most effective, achieving an impressive 98.86% accuracy on the development set.