SLP: Native Language Identification Challenge

Group 23
Daniele Avolio ist1111559, Eduardo Rodrigues ist1111684

1. Introduction

Speech recognition and classification have become essential technologies in human-computer interaction, offering natural and intuitive interfaces for various devices and applications. This study focuses on simulating a native language identification challenge, where participants receive training, development, and evaluation datasets, alongside baseline systems for closed-set identification of the native language of foreign English speakers. The target languages include Chinese, German, Hindi, and Italian.

The objective is to develop the most effective native language identification system. During the initial phase, participants are required to understand and enhance a baseline system utilizing MFCC features and GMM models. In the subsequent phase, they will explore modern systems based on self-supervised learning. The project encourages innovation through the modification and combination of different systems to achieve optimal performance. Preliminary results demonstrate significant improvements in accuracy, showcasing the potential of advanced techniques in enhancing native language identification.

2. Classical models based on conventional features

This section describes the basic system, which consists of MFCC feature extraction followed by GMM classification. MFCC feature extraction includes optional components such as Delta and Double-delta calculation, Shifted Delta Cepstrum (SDC), Voice Activity Detection (VAD) and Cepstral mean and variance normalisation (CMVN).

2.1. MFCC Feature Extraction

At the heart of our baseline system lies the Mel-Frequency Cepstral Coefficients (MFCC) feature extraction process. The **feat_extract** function encapsulates this process, providing a flexible framework for extracting MFCC features from audio files. Users can customize various parameters such as the number of MFCC coefficients, delta computation, Shifted Delta Cepstrum (SDC), Voice Activity Detection (VAD), and Cepstral Mean Variance Normalization (CMVN) based on their requirements.

2.2. Optional methods for MFCC feature extraction

The optional methods for MFCC feature extraction include computing deltas, Shifted Delta Cepstrum (SDC), Voice Activity Detection (VAD), and Cepstral Mean Variance Normalization (CMVN). These methods enhance the quality of the extracted features and improve the performance of the language identification system. However, we didn't actually get very good results so probably there is something wrong with the implementation, but we couldn't manage to find what could have been wrong. Surely, we tested some combination of methods, and the most promising one was the VAD, followed by the CMVN.

2.3. Model Training

We employ a supervised learning approach to train language identification models using Gaussian Mixture Models (GMMs). For each native language in the dataset, we train a separate GMM model using the extracted features. The models learn the underlying distributions of features specific to each language and use this information to classify new audio samples. The training process was quite fast, and we could train the models in a few minutes. We used the **train100** datasets to train the models, and we used the **dev** dataset, of course, to evaluate them. We even tried to use the **train** datasets, but it was taking too long to run some tests on it. On the *System Optimization* section we will talk more about the training process.

2.4. Evaluation

To assess the performance of the language identification system, we evaluate it on a development set. This set contains audio samples with known ground truth labels. We measure metrics such as accuracy, precision, recall, and F1-score to quantify the system's performance across different languages. However, the result were not satisfactory, achieving a score of roughly 0.6 in the Kaggle competition. We believe that the problem was in the feature extraction, but we couldn't manage to find what was wrong. We tried to use different combinations of methods, but the results were always the same. We also tried to use the **train** datasets, but even with it the results were kind of the same, so probably the problem was in the implementation.

2.5. System Optimization

Based on the evaluation results, we fine-tune the system parameters and explore techniques for improving classification accuracy. This may involve adjusting the number of Gaussian components in the GMMs, optimizing feature extraction parameters, or experimenting with alternative machine learning algorithms. In particular, we tried to estimate the best number of components for the GMMs using the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC). The process was very slow, so we did it only the first time, and then we used the best number of components for the following experiments.

3. Native Language Identification with Pre-trained Models

In this section, we explore two approaches to NLI using pre-trained models: x-vector based and self-supervised learning (SSL) based methods. This is based on the second part of the project, where we were guided to use the s3prl toolkit to implement the SSL approach and to use pretrained speaker embeddings with the x-vector based approach. We will discuss the results we got from the x-vector based approach and the problems we had with the SSL approach.

3.1. X-vector Based Approach

X-vectors, derived from deep neural networks trained for speaker identification, have emerged as powerful embeddings for various speech processing tasks. In this section, we delve into the process of extracting x-vectors from audio data using pre-trained models. We demonstrate how to train a simple linear SVM classifier on top of these embeddings and evaluate its performance on a development set. Additionally, we experiment with alternative models like **RandomForest, GradientBoosting, Logistic Regression, and KNN**. The x-vector based approach yielded promising results on the development set.

	Precision	**Recall**	**F1-Score**	**Support**
CHI	0.76	0.87	0.81	39
GER	0.72	0.64	0.67	44
HIN	0.96	0.94	0.95	47
ITA	0.70	0.70	0.70	46
Accuracy			0.78	176
Macro Avg	0.78	0.78	0.78	176
Weighted Avg	0.78	0.78	0.78	176

Table 1: *Precision, Recall, F1-Score and Support for different languages*

The best model was the Logistic Regression, with an overall score of 0.87. Compared to the GMM model, the x-vector based approach achieved a significant improvement in accuracy, precision, recall, and F1-score metrics. Note that this was the best result obtained using the transformation function **spkrec-ecapa-voxceleb** that should perform worse than **lang-id-voxlingua107-ecapa**. We believe that the results could be improved by using the **train** datasets, but the transformation function **lang-id-voxlingua107-ecapa** was taking too long to run.

3.2. Self-Supervised Learning (SSL) Approach

The s3prl toolkit offers a suite of self-supervised pre-trained models for speech processing tasks. In this section, we explore how to leverage SSL models for NLI by fine-tuning them on our dataset. We discuss the downstream task setup and the architecture of the simple model used for NLI. Furthermore, we provide instructions for running scripts to utilize SSL models and save results for evaluation. The SSL approach using the s3prl toolkit provided competitive results:

However we had some problems to run this, because I couldn't get this to work on Windows (I tried to run it on WSL, but it didn't work) and I don't know why. Unfortunately, I didn't have time to try to do more tests so we just kept the results we got from the lab when we did manage to run it throug Eduardo's computer. Since it's a very powerful tool, we believe that the results could be much better than the ones we got from the x-vector based approach if we had more time to work on it.

Table 2: *Language Classification Metrics*

Language	Precision (%)	Recall (%)	F1-score (%)
CHI	76	87	81
GER	72	64	67
HIN	96	94	95
ITA	70	70	70
Overall			78.41

4. Conclusion

The conclusion of this study highlights the evolution and diversity of approaches to native language identification, which are essential for a wide range of applications in human-computer interaction, language processing, and security.

Classical models, based on conventional features such as mel frequency cepstral coefficients (MFCC) and Gaussian mixture models (GMM), lay a solid foundation for native language identification. However, it's clear that new approaches like x-vectors and self-supervised learning (SSL) models offer significantly better performance, as demonstrated by the results obtained in this project.

Preliminary results reveal that the use of x-vectors and SSL models results in promising precision, recall and F1-score metrics, effectively competing with classical models in several target languages.

It's clear that the future of native language identification lies in the exploration and integration of advanced techniques like x-vectors and SSL models, but it's also important to consider the computational resources and expertise required to implement these approaches effectively. Actually, the implementation being guided made the process easier and it would have been much harder to do it manually and in this short amount of time.

A very important note is that we did all our tests only on the **train100** datasets due to time constraints. We believe that the results could be improved by using the **train** datasets, but it would have taken too long to run the tests. Maybe finding another transformation function different from the **2** we used could have improved the results as well, going more specifically to the languages we were trying to identify.