Lab 2 SLP 2022/2023, Group 3

June 4, 2023

Abstract

Language Identification (LI) is one of the main tasks in speech processing. In this work, we will explore two approaches to this task: one based on GMM and another on *x-vectors*.

1. Experimental Setup

We are performing **speech pattern classification**, specifically, we propose a data-driven solution to an **identification** problem. The data is a set of audio files containing speech of only one of the 6 target languages. The dataset is organized in 3 partitions: TRAIN100 - the training set, DEV - the development set, EVL - the test (evaluation) set. [1]

The dataset is somewhat balanced: all languages have the same number of files (100), and the difference between the one with the most (English) and least (Portuguese) data is around 11 minutes. This is particularly important, as the decision could otherwise become more skewed towards classes that are more represented in the training data.

We designed two architectures to solve this problem, which we will explain in more detail in the next section:

- An improved baseline, i.e., a classical speech pattern classification approach. In this track, we emphasized the **feature extraction process** rather than the choice of the model (given the enormous preponderance of GMM in traditional approaches to tasks of this type).
- A pre-trained-based model, that allowed us to learn from an embedding representation of the waveforms. In this track, we focused more on **model architectural choices**, building upon the representational power of the pre-trained embeddings.

2. Approaches

2.1. The improved baseline (Track 1)

Our baseline model was based on classical machine learning techniques. Here, our main focus was to extract the **language discriminative features from the waveform**, removing speaker and noise-related characteristics.

Our feature extraction process was guided by [1] and [2]. We used local features (at the frame level), that mostly captured spectral (e.g., MFCCs) and prosodic (e.g., the energy used to compute VAD) information.

- We used **MFCC**, which are a set of perceptually based features. For a simple task like LI, only the first 7 are sufficient (more details concerning this on Section 3). The 1st coefficient was discard due to correlation with the energy. We used a sampling rate of 16000, a window size of 512 for the FFT calculation and a hop-length of 160 frames.
- On top of MFCCs, we used the **SDC**, which is an improvement of the deltas (velocity and acceleration) due to its *ability to incorporate additional temporal information, spanning multiple frames, into the feature vector* [3]. We considered a distance between vectors $D = 1$ for the delta computation, a distance between blocks $P = 3$, and considered $K = 7$ consecutive blocks for the SDC construction.
- Moreover, we reduced the data size by using **VAD**. This step is important because silence may bias models, and we want to purely focus on features from speech. For that purpose, we computed each audio's Root Mean Square using the same hop length and frame size employed in the computation of the MFFC's. We then proceeded to fit a 2-component GMM on these energies and discarded frames that were assigned to the centroid associated with the lowest energy.
- Finally, we applied **CMVN** to remove the channel effect. Given the homomorphic property of cepstral representations, we are then able to deconvolve the channel effect (assumed to be constant) from the input speech signal through mean and variance normalization.

Once the most important part was completed, we applied a **GMM** with 256 gaussians, which is the classical approach in speech pattern classification problems.

2.2. The advanced approach (Track 2)

On the second part of the work, we resorted to 256-dimensional *x-vectors* extracted from the VoxLingua107 ECAPA-TDNN Spoken Language Identification Model [4]. This model was also trained for a Language Identification task, albeit for a bigger universe of languages (107). The use of this pretrained model simultaneously allows the simplification of the feature extraction process and leverages the application of more expressive models. We used the accuracy on the dev set as a criterion for the choice of our model.

Firstly, we tried out SVM's with linear, gaussian and 2nd and 3rd order polynomial kernels. All models yielded accuracies above 94%, with the linear kernel SVM scoring the highest. We also experimented with a simple logistic regression classifier, having obtained a result in the same range. Furthermore, we tested the use of feedforward neural networks, having performed grid search on the number of hidden layers and hidden unites per layer on the set $(n_{\text{layers}}, n_{\text{units}}) \in \{20, 30, 40, 50, 60, 70, 80, 90, 100\} \times \{1, 2, 3, 4, 5\}$. Since the accuracies were all near 100% for the train set and between 70% and 80% for the dev set, this last model seemed to suffer from overfitting[2]. To tackle that problem, we employed PCA, having thereby observed that 35 components were able to retain 95%

[1] We didn't train our model using the complete train set because we didn't have enough computational resources

[2] This may have been due to the different nature between these sets (News and YouTube videos). The trained model may have captured some characteristrics specific to the training conditions that aren't seen on the dev set

of the dataset's total variance. Nevertheless, we did not observe any significant accuracy improvements.

Another approach that we employed was the computation of *x-vectors* for 2 second audio segments of each audio file, giving rise to $\sim$14000 training instances. At test time, the predicted class of each audio was computed through a majority vote of the classes assigned to each *x-vector*. Performing a similar grid search as before, we obtain accuracies around 87% for the feed-forward network. Moreover, we tried an ensemble approach by training 10 SVM/logistic regression classifiers on disjoint data partitions to try to improve the previous results, having observed a performance decrease in that case.

Our last strategy in this stage consisted of considering the likelihoods outputed by the pretrained model as our input feature vectors. Having applied the same classifiers as before (logistic regression, SVM with different kernels and FFN with different number of hidden layers and hidden units), we noticed an increase of the FFN's performance to 90%, while the other classifier's accuracies registered decreases. Thus, we chose a **SVM with a linear kernel on top of x-vectors computed over the entirety of each audio file** as our proposed classifier.

3. Results and Discussion

The results of the first architecture can be found in Figure 1. Note that this is the result of the best configuration found:

- To decide on the best set of features, we assumed independence between them and decided the best setting incrementally (e.g., using no deltas was better than using 1 or 2, so we evaluated whether to use SDC having fixed 0 deltas).
- To decide on the hyperparameters for the GMM, we compared the accuracy (on the dev set) obtained for $\{16, 32, 64, 128, 256\}$.

Furthermore, we trained this exact configuration using 13 MFCCs and our performance went down to 53%. This shows that, for a simple task like ours, adding more complexity will result in overfitting (learning features that aren't language discriminative). As an alternative, we also tried a UBM-based architecture, following the one presented in [5]. We tried both mean MAP adaptation and mean, covariance and weight MAP adaptation, but we registered a decrease of 10% in the accuracy.

The results of the final architecture are in Figure 2. The discussion of the best architecture was done above, in 2.2.

In both confusion matrices, each number indicates a language in the order {Basque, Catalan, English, Galician, Portuguese, Spanish}. For example, in Figure 1, cell $(2, 3)$ means that 1 sample that was of English was misclassified as Galician.

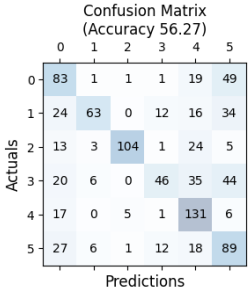

Figure 1: *Confusion matrix and overall performance of the baseline model.*

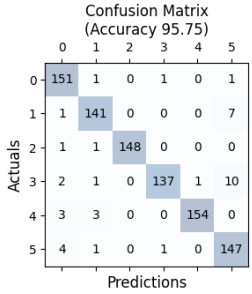

Figure 2: *Confusion matrix and overall performance of the final (advanced) model.*

4. Concluding Remarks and Future Work

Regarding the baseline model, we were only able to capture some language discriminative features of the speech signal. This may have been due to several reasons: the different nature of the train and the dev set, and the small data used to train, which both may have led to overfitting.

Given more time, we would test the inclusion of other features: speech enhancement, like the Wiener filter and spectral subtraction, and photactic information, as [1] mentions, along with acoustic information, is useful in LI.

After experimenting with *x-vectors*, we conclude that these embeddings have a remarkable representational power, as linear classifiers surprisingly generate the best results. Nonetheless, we note that there seems to be an irreducible error, which may be attributed to the high similarity between languages spoken in Spain (incidentally, the biggest source of error in the dev set comes from the misclassification of Galician audios as Spanish).

Furthermore, we postulate that the various convolutions present in the pre-trained model are more capable of capturing long range patterns in audios in comparison to the applied majority voting strategy, hence the superior accuracies obtained by using one *x-vector* per audio file.

For future work, we suggest fine-tuning the pretrained model by training the last FFN; specifically, keep the existing weights after discarding the connections to output nodes that don't correspond to any of the 6 languages we want to identify.

5. References

[1] D. Deshwal, P. Sangwan, and D. Kumar, "Feature extraction methods in language identification: a survey," *Wireless Personal Communications*, vol. 107, pp. 2071–2103, 2019.

[2] A. Abad, "Speech pattern classification," fenix.tecnico.ulisboa.pt/downloadFile/1970943312402524/PF2023v2.pdf, accessed: 2023-05-24.

[3] F. Allen, E. Ambikairajah, and J. Epps, "Language identification using warping and the shifted delta cepstrum," in *2005 IEEE 7th Workshop on Multimedia Signal Processing*. IEEE, 2005, pp. 1–4.

[4] "Voxlingua107 ecapa-tdnn spoken language identification model," https://huggingface.co/speechbrain/lang-id-voxlingua107-ecapa, accessed: 2023-06-03.

[5] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.