Intent Detection Audio Classification Problem

Luca Agnese
Politecnico di Torino
Student id: s303382
l.agnese@studenti.polito.it

Flavio Spuri
Politecnico di Torino
Student id: s303657
s303657@studenti.polito.it

***Abstract*—In this report, we discuss a possible solution to an intent detection problem. In particular, given a dataset consisting of a collection of recordings with some additional features, we want to classify the expressed intent of the speaker. In the proposed approach we extract different types of audio features and summarize them by dividing them into chunks and computing some statistics on them. This approach significantly outperforms the given baseline in terms of accuracy score.**

I. Problem Overview

The aim of this work was to predict the intent expressed in an audio recording. In particular, the intent is composed of two parts: an *action* (e.g. ”increase”) and an *object* (e.g. ”volume”).
The given dataset is split into two portions:

- A *development* set consisting of 9854 labelled samples.
- An *evaluation* set consisting of 1455 unlabelled samples.

Each sample is characterized by several attributes:

- *path*: the path of the audio file.
- *speakerId*: the id of the speaker.
- *Self-reported-fluency-level*: the speaking fluency of the speaker.
- *First language spoken*: the first language spoken by the speaker.
- *Current language used for work/school*: the main language spoken by the speaker during daily activities.
- *gender*: the gender of the speaker.
- *ageRange*: the age range of the speaker.

In the development set, we are also given the *action* and *object*, whose combination gives the intent. The only exception is for the *changelanguage* intent, for which the relative recordings have an empty value in the *object* column and in the *action* column an erroneous value of ”change language”.

Let’s make some considerations based on the development set: first, the given dataset is not very well balanced in the target feature as we can observe in Figure 1. Furthermore, while there is not any missing value for any feature, all of them are very unbalanced with the only exception of *gender*, as can be observed in Figure 2. Notice that in this discussion we are purposely omitting the *speakerId* and *path* fields, as the former doesn’t really contain any relevant information and the latter is used to identify the audio recording associated to the data point.

For what regards the recordings, we can first observe that they all have an original sample rate of 16 KHz (with the

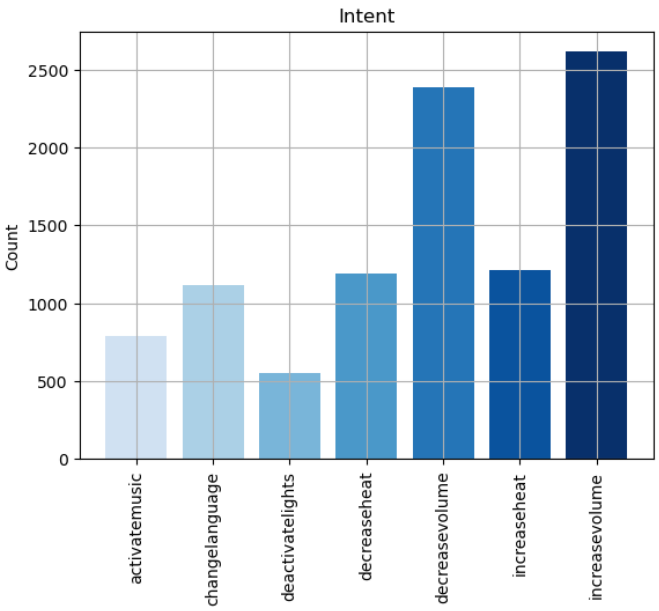


Fig. 1. Number of occurrences per target values.

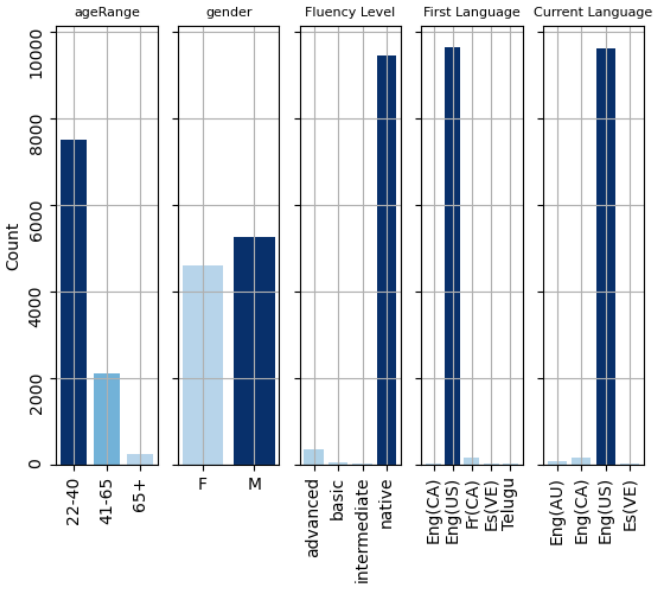


Fig. 2. Number of occurrences per feature value.

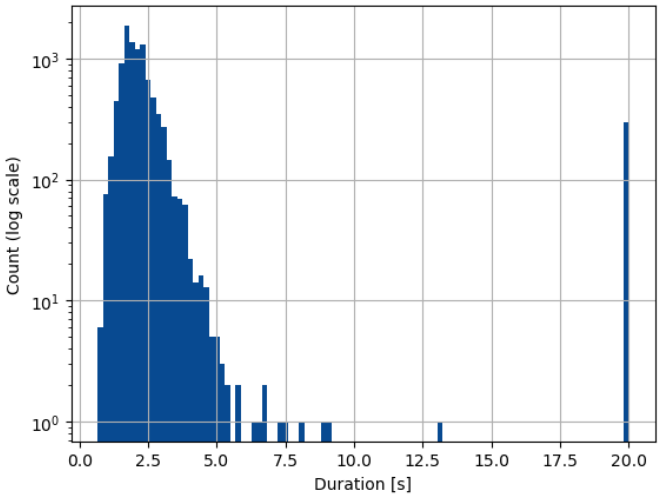


Fig. 3. Distribution of the signals' durations.

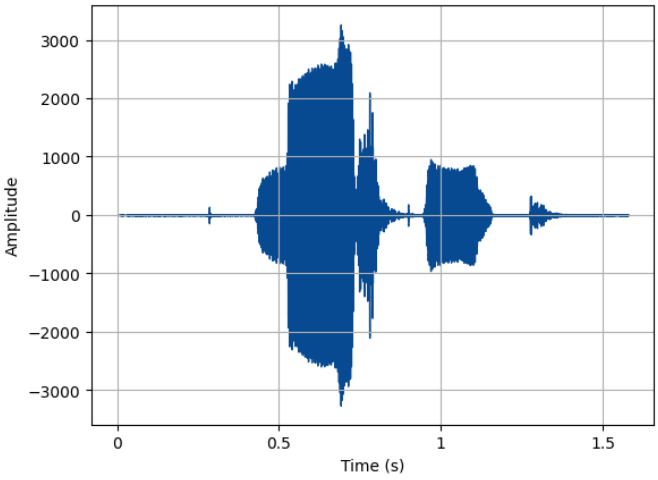


Fig. 4. Representation of a recording in the time domain

exception of a single recording). While the sampling rate is the same throughout all the recordings, their duration varies: by looking at the distribution of the durations in Figure 3, we can see that there are some outliers deviating from the mean duration of 2.64 *s*. As the presence of these outliers cannot be reduced to a single cause we will further discuss it in the preprocessing section. Having recordings of different durations also meant that we needed a strategy to extract a uniform amount of features from each of them.

Visualizing several audio samples both in the time domain and in the frequency domain helped us understand some characteristics of such signals. Firstly, as can be seen in Figure 4, many signals present leading and trailing parts of silence. Since the recording have a bit depth equal to 16 bits, we know that the amplitudes' values are bound to be within the range [-32768, 32767]. For what regards the frequency domain, as it is common in this kind of application, we deemed the decibel (thus logarithmic) scale to be more significant then a linear one, hence for the rest of our discussion we will assume to be using the former, unless otherwise stated.

II. Proposed approach

A. Preprocessing

Regarding the "non-audio" features we expect them to possibly retain some useful knowledge about the signals, as they contain information about the accent (e.g. "First Language spoken") or the voice pitch (e.g. "ageRange") of the speaker. However, as we noticed before, most of them are too unbalanced to be used. Thus, we decided to drop them, keeping only the *gender* feature. We preprocessed this feature by converting it to a binary variable. Finally, we adjust the misrepresented label of *changelanguage*.

Focusing now on the audio side, we first did some cleaning steps on the given recordings. In particular, given a signal we reduced the noise and then trimmed the trailing and leading silence regions. The various steps of the transformation of a signal can be seen in Figure 5. After these steps, most of the recordings with a duration deviating too much from the mean had their duration reduced to values in line with the overall distribution; for the remaining outliers, upon manual inspection, we observed that they either were extremely noisy recordings or they contained mistakes from the speakers, thus we decided to drop them. The distribution of the durations after the cleaning steps is shown in Figure 6.

We then identified different quantities to be computed on top of the signals. First, as it is common in speech recognition tasks, we considered the **mel spectrograms** of the signals which retain information of both the time and frequency domains. Then, we also considered the **MFCCs** (*Mel-Frequency Cepstral Coefficients*) which are commonly used in speech recognition tasks and they are computed from the FFT power coefficients [1], which again retain information in both domains. A visual representation of the mel spectrogram and the MFCCs of a signal is shown in Figure 7, where we have the time on the x-axis, the frequency and coefficients respectively on the y-axis and the amplitude as a third dimension codified through colors. Finally, we considered other two quantities related to the time representation of the signals: the **ZCR** (*Zero-Crossing Rate*) and the **RMS** (*Root Mean Square energy*) which respectively represent the rate at which the signal alternates between positive and negative [2] and an approximation of the loudness of the signal. The mel spectrograms and the MFCCs can be interpreted as $N_{spect} \times M$ $N_{mfcc} \times M$ matrices respectively, while the ZCR and the RMS are represented with M-dimensional row vectors, where:

- M is a quantity proportional to the length of the signals. It is the same for all the quantities when computed on the same signals (using equal-sized windows and hops in the FFTs).
- N_{mfcc} is equal to the number of coefficients considered.
- N_{spect} is equal to the number of Mel bands used in the computation of the mel spectrogram.

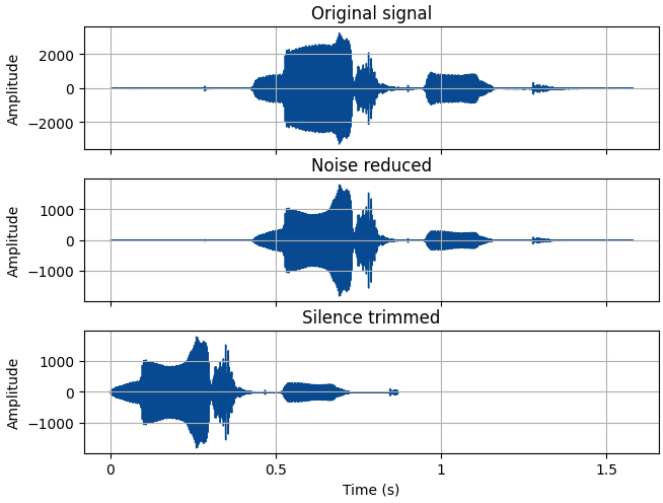


Fig. 5. Signal at various stages of the cleaning process

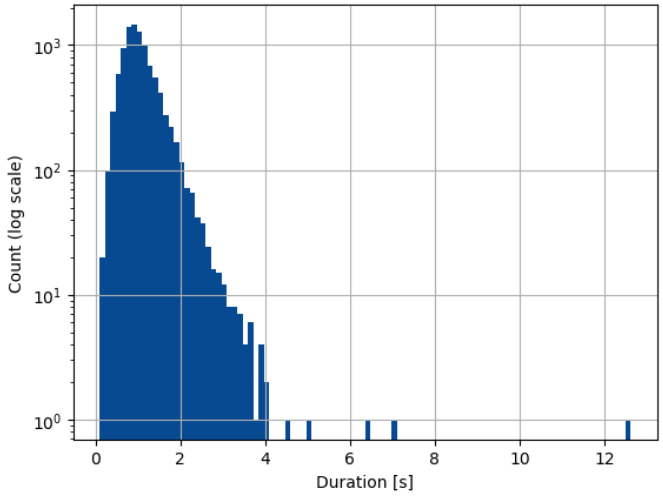
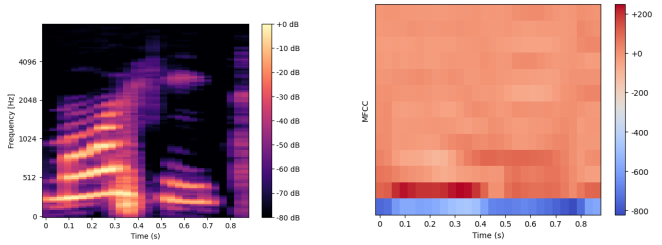


Fig. 6. Distribution of the signals' durations after cleaning

Thus, in order to extract a uniform amount of features we decided to split such matrices and vectors into blocks and summarizing each block by computing some statistics on it. In particular, we considered the mean, standard deviation, minimum and maximum of each block of the mel spectrograms and MFCCs and the mean and standard deviation for ZCR and RMS. Since MFCCs are built upon mel spectrograms, one can expect to have a high correlation between the features extracted from the two. Indeed, as can be seen in Figure 8 the correlation matrices show that there is a strong correlation between the statistics extracted from MFCCs and spectrograms, hence we eventually decided to not use mel spectrograms in our approach.

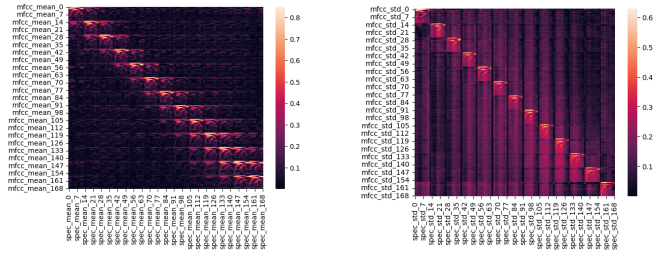
B. Model selection

Since an approach that uses different combinations of the aforementioned features has been shown to perform the best



(a) Mel Spectrogram (b) MFCC

Fig. 7. MFCC and Mel Spectrogram visualizations for an audio sample



(a) Mean (b) Std

Fig. 8. Matrices of the absolute values of the correlation between mean of MFCCs and mel spectrograms and standard deviation of MFCCs and mel spectrograms

when all of them are used together [3], although basing the study on deep learning models, we tried to reproduce the same results by using standard machine learning algorithms. In particular, we decided to use **SVCs** (*Support Vector Classifiers*) and **RFCs** (*Random Forest Classifiers*) since they have been shown to work well in audio classification tasks [4] [5]. In the case of the SVC, we first applied a normalization step, using a z-score normalization.

C. Hyperparameters tuning

To select the best hyperparameters configuration, a grid search using an 80/20 train/test split has been run. In particular, we need to consider two types of hyperparameters:

- The ones referring to the SVCs and RFCs.
- The one referring to the number of blocks to be computed.

With regard to the latter, we decided to keep the number of splits n on both axes the same and for all the quantities the same, in order to significantly reduce the number of possible combinations; in other words, we split the mel spectrograms' and MFCCs' matrices in $n \times n$ blocks, and the ZCRs' and RMSs' vectors in n blocks. In particular, we could perform at most 13 splits for each axis as we used a number of MFCCs equal to 13, which is commonly used in the task of speech recognition. On each split, both a grid search for the RFC and the SVC has been run, considering the parameters that can be seen in Table I. Performances have been assessed by means of the accuracy score.

Model	Parameter	Values
SVC	*n*	{1, 4, 7, 10, 13}
	C	{1,5,10,50,500,1000}
	Kernel	{rbf, sigmoid}
	Gamma	{scale, auto}
RFC	*n*	{1, 4, 7, 10, 13}
	n_estimators	1000
	criterion	{Gini, entropy}
	max_features	{sqrt, log2}

TABLE I
HYPERPARAMETERS VALUES CONSIDERED FOR THE GRID SEARCH

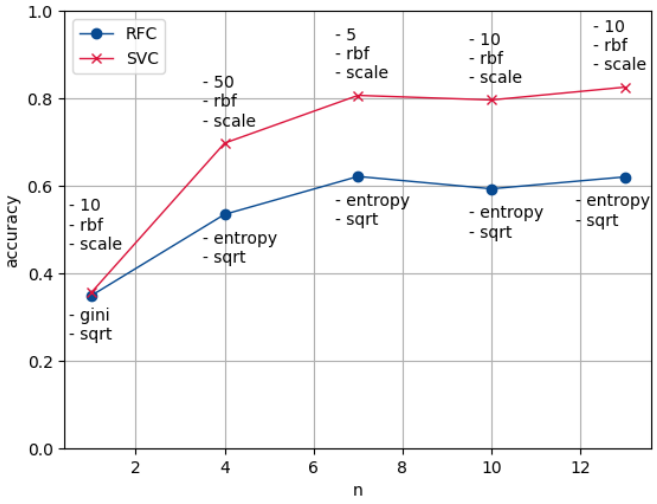


Fig. 9. Performance of the best configuration of the classifiers for each split

III. RESULTS

As can be seen in Figure 9 from the results of the grid searches it is immediately clear that the SVCs outperform the RFCs in this task. Overall, the best configuration has been found for:

- *Model*: SVC
- *n*: 13
- *C*: 10
- *Kernel*: "rbf"
- *Gamma*: "scale"

with an accuracy on the test set of 0.8255.

After obtaining the best hyperparameters configuration, the model has been trained on the whole development set and has been used to label the intent on the evaluation set, obtaining a public score of 0.76.

IV. DISCUSSION

The proposed approach obtains a result significantly higher than the given baseline of 0.334. However, there is a not negligible loss of accuracy from the one computed on the test set and the one computed for the public score.

The following aspects may be considered in order to further improve the results:

- Additional features may be extracted. One possibility would be to consider additional statistics for each block; in particular, extracting higher order statistics (e.g. *skewness* and *kurtosis*) may lead to a more robust system. Notice that such an addition would increase the number of features by n^2 for each additional statistics computed on the MFCCs' blocks, thus the computational cost would become much higher.
- A more complex grid search could be leveraged. In particular, it would be possible to both decouple the number of splits along the axes and decouple the number of splits between different audio features.
- A big improvement would most likely be obtained by using deep learning models. In particular, from the literature the CNNs seem to be a common choice for speech recognition tasks [6].

REFERENCES

[1] Jianfei Cai. Liang-Tien Chia. Changsheng Xu. Qi Tian. Min Xu, Ling-Yu Duan, "HMM-based audio keyword generation," *LNCS 3333*, pp. 566–574, 2004.

[2] C. H. Chen, *Signal Processing Handbook*. Marcel Dekker,Inc, 1988.

[3] M. B. T. S. Muhammad Turab, Teerath Kumar, "Investigating multi-feature selection and ensembling for audio classification," 2022.

[4] S. P. P. Dhanalakshmi and V. Ramalingam, "Classification of audio signals using svm and rbfnn," *Expert Systems with Applications*, pp. 6069–6075, 2009.

[5] G. D. Greenwade, "Smartphone-based real-time classification of noise signals using subband features and random forest classifier," *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 2204–2208, 2016.

[6] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.