Pitch-shifting based data-augmentation for Intent recognition with MFCC

Marcelo Bastos Ferreira, Gustavo Nicoletti Rosa
Politecnico di Torino
Student ids: s308964, s317672
marcelo.bastosferreira@studenti.polito.it, gustavo.nicolettirosa@studenti.polito.it

***Abstract*—In this report, we present a speech intent detection algorithm using various preprocessing techniques, including data augmentation, pitch extraction, Mel spectrogram, and delta features. We evaluated two different classification models Random Forest (RF) and Support Vector Machine (SVM). The results of the experiments showed that the best algorithm achieved an accuracy score of 0.95 for the classification of multiple intents.**

I. Problem overview

Intent detection (ID) from audio is a Natural Language Processing (NLP) task where the model must understand the intention from utterances. It goes beyond speech recognition because apart from understanding the phrases, the model must understand the context and the way of speaking, which may differ for each speaker by varying gender, age, ethnicity, and personal physiology. We have used a reduced version of the Fluent Speech Commands dataset to train and test a system able to recognize a set of spoken commands to interact with a typical voice assistant in a smart home scenario with various different wordings. It is composed of audio files, a few pieces of information about the speaker, and the intention of the utterance separated into action and object.

We have had the dataset separated into:

- a *development set*, with 9854 utterances
- an *evaluation set*, with 1455 utterances

We used the first set to train the model and the second to test our model through the accuracy score. In the development set we have speakers characterized by the ages in the ranges 22-40, 41-65, and 65+; by a fairly even distribution of males and females; people from the United States of America (USA), Canada, Australia, and Venezuela; whose first languages are either English (USA), Canadian English, Telugu, or Venezuelan Spanish; and English fluency level are divided in basic, intermediate, advanced and native. Excluding outliers, the audios were on average 2,617s long. Manually inspecting the longer audios, their duration is due to a long time of silence being recorded.

The intention classes are not well distributed in the *evaluation set* (Fig.1), for it has more intentions related to the volume of the device than the other classes, however, in the application of a voice assistant it is expected to have more requests of this type because the volume is very dynamic, *e.g.* a user would normally change the volume multiple times during a single media play.

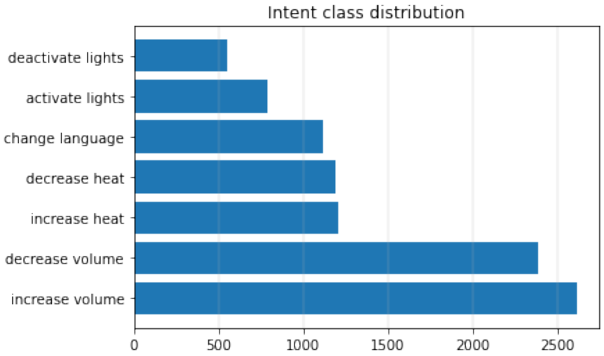

Fig. 1. Intent distribution in *development set*

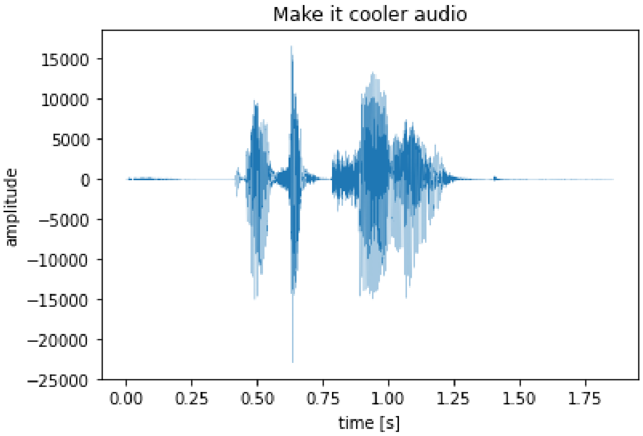

Fig. 2. Make it cooler audio in time domain before trimming

Fig.2 plots the audio of the utterance ”Make it cooler”, it represents the type of data we are dealing with. Another way of representing audios is through their Fourier transform, where we can study the energy distribution for each frequency (Fig. 3). Vowels primarily lie in the range 250 – 2,000 Hz, voiced consonants in the range 250 – 4,000 Hz, and unvoiced consonants vary considerably in the frequency range 2,000 – 8,000 Hz. Most audios in the development set are sampled in 16 kHz, respecting the Nyquist theorem to allow a complete sampling of all English phonemes. And few audios are sampled in 22 kHz. The pitch, besides varying from person to person [1], also changes for the same person, influenced by

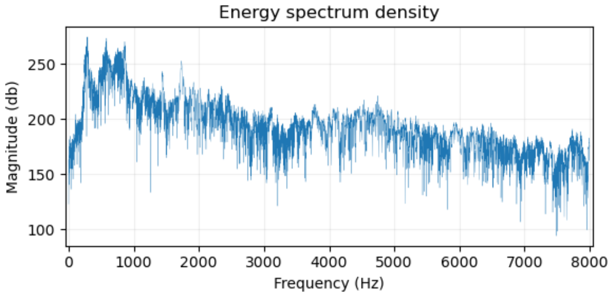

Fig. 3. Make it cooler audio in frequency domain

the emotions they convey [2], so in an intention recognition problem, this feature is also important for our model.

II. Proposed approach

A. *Preprocessing*

Analyzing the data, it is possible to notice that very few audios come from non-native speakers of American English, or from people above the age of 65, so all those *could* be treated as outliers and removed, as discussed later on. Also, by inspection, we have noticed that many of these ”outliers” were incomprehensible. However, including them in the model training improved the results. Next, all audios having a sample rate equal to 22050 Hz were sample-rate converted to their equivalent audio having a sample rate of 16 kHz.

As previously noted in Problem Overview, some audios have long periods of silence. The distribution of the length of the audios before our preprocessing can be seen in Fig. 4, with an average of 2.63s. So for all audio signals, we have first hard-trimmed at 4 seconds, then trimmed leading and trailing silence, where the threshold to be considered silence is to be more than 20 dB under the max amplitude of the signal (the threshold was treated as a hyperparameter, as discussed also later on).

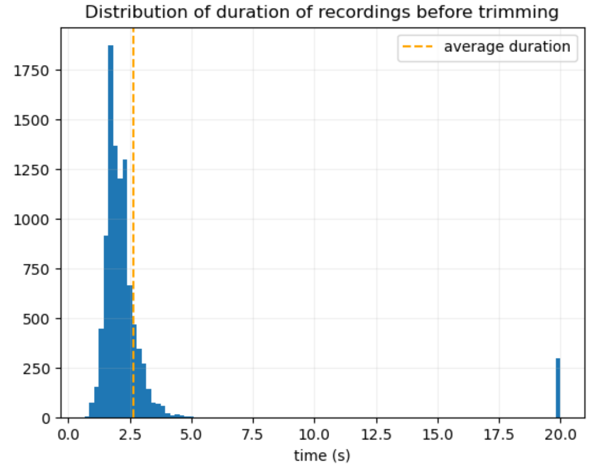

Fig. 4. Histogram of the duration of audios before trimming

Most of the models only accept samples of the same number of attributes, and to deal with the relative speed of speech for each person we have decided to time stretch all of the audios [3], without altering the frequency properties, to the length of 21 thousand samples, equivalent to 1,3125 seconds of audio. Although the average time of the audios after trimming was 1.078s, we decided that stretching most of the audios could be more beneficial to the model than shrinking the longer ones, as the opposite could lead to having a lower amount of information for the audios on average.

Multiple papers show that by increasing the number of samples by augmenting the data, the model improves in generalization and accuracy [4]. For that reason, for each audio that a male speaker recorded, three new audio samples were created with pitch shifting by 2, 4, and 6 semitones higher to simulate a woman speaking and to create a common ground between the male and female pitch ranges [1]. For female speakers' recordings, the opposite was done. In both models, we have added as features the original length of the audio after trimming but before time stretching, its maximum amplitude, and the pitch contour after both time stretching and pitch shifting, since they convey important information on the utterance itself. We also extracted other frequency-based features for our models, including Mel-Frequency Cepstral Coefficients (MFCCs), delta features, and Mel spectrograms.

The spectrogram (Fig. 5) is a time-frequency representation of a signal, where the original time domain signal is split into chunks (windows, that can be overlapped). Each chunk is then Fourier Transformed and appended horizontally (as a column vector) to the other Fourier-transformed chunks. To transform a spectrogram into a melody-scaled frequency spectrogram, just a re-scale in the frequencies using $m = 2595\log_{10}(1 + \frac{f}{700})$ is needed. It is used sometimes in speech recognition algorithms because studies have shown that humans do not perceive frequencies on a linear scale. The Mel-frequency cepstral coefficients (MFCC) take it a step further (Fig. 6). To get the coefficients from the *linear* spectrogram, a few steps are needed:

1) Mel-scale the columns of the linear scaled spectrogram;
2) Apply a log transformation using an arbitrary window, the triangular window is usually applied;
3) Perform a discrete cosine transform on the resulting vectors;

On the other hand, the delta (Fig. 6) and the delta-delta features are, respectively, the estimate of the derivative in time of the matrix made of the MFCC of first and second order [5].

As almost all preprocessing techniques were treated as ”hyperparameters”, testing and analyzing if it is worth applying, and how to apply such techniques, take into consideration that not necessarily the following steps are included in the final, best-performing model:

1) Removal of the ”outliers” specified at the beginning of this section;
2) Normalization of every audio with respect to their maximum amplitude;

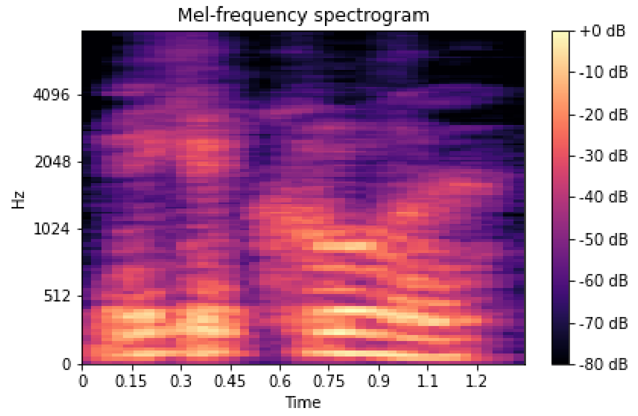

Fig. 5. Mel-scaled frequency spectrogram for the preprocessed "Make it cooler" utterance

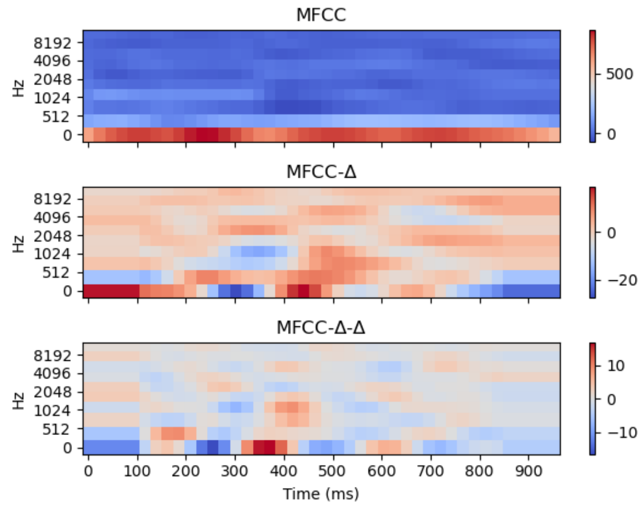

Fig. 6. MFCCs and delta features for the preprocessed "Make it cooler" utterance

3) Pooling algorithm (max or mean) on the Mel-spectrogram or the Mel-frequency cepstrum;
4) Using delta and delta-delta features, together with the one-hot encoded age range, gender, first language spoken and current language used for school/work, and the order encoded self-reported fluency level;
5) Further normalization with the z-score so that all features have the same range of values, to avoid giving importance to features wrongly, therefore improving the performance of the hyperplane subdivisions (in the case of the SVC model).

B. Model selection

We opted for end-to-end speech recognition techniques [6], without going through a text phase, because it is the current state-of-the-art and because we would need a wide training data to go from audio to text, and only then apply our intent recognition model.

a) Random Forest: is an ensemble learning technique with a collection of Decision Trees where each tree trains on a subset of features and usually trains with a subset of the samples. Then the classification is resultant of a majority voting among the trees. This technique is used to avoid overfitting based on specific features.

b) Support Vector Machine: creates a non-linear feature vector space by taking advantage of a kernel function when performing the linear maximization of the margin distance in such space among different classes [7], creating a non-linear hyperplane in the original space.

C. Hyperparameters tuning

Many different things were experimented on, including:

a) Preprocessing: In addition to the steps mentioned in the previous subsection, we also tried:

- zero padding audios to make them of equal length instead of time stretching [3];
- pitch normalizing the audios [8];
- after pitch shifting data augmenting [4], compute pitch contour with library-implemented function or by the direct formula (Eq. 1), where N is the shift in semitones inside an octave of 12 semitones;
- decibel threshold to consider something as silence;
- window length, hop length, and band-pass filter frequencies used for the Mel-spectrogram, MFCCs, delta, and delta-delta features;
- the number of coefficients, in the case of the MFCCs and derived features;

$$pitch_{new} = pitch_{original} * 2^{N/12} \tag{1}$$

b) Random Forest: For this classifier, we have evaluated the cross-validation by testing combinations of preprocessing steps and hyperparameters with a thousand as the fixed number of estimators (Decision trees) because we used over 1300 features. The hyperparameters of the Random Forest that were tuned include were:

Quality of split criterion	Gini, Entropy
Minimum samples on split	2, 5, 10
Features considered on split	$\sqrt{\#features)}$, $\log_2(\#features)$
Bootstrap	Yes, No
Class weight	Class frequency, No

c) SVC: Since this classifier demonstrated a higher potential in our tests we have tuned the hyperparameters with a more extensive grid search process. For this model, the hyperparameters that were tuned include the regularization parameter (C) and their class weights, the kernel function, and the influence of a single training sample (γ). Specifically:

Regularization parameter C	0.1, 0.5, 1, 2, 5
Regularization weights	class balanced, unweighted
Sample influence γ	0.005, $1/(\#features)$, $1/(\#features * X_{var})$
Kernel function	Polynomial, Sigmoid, Radial basis

III. Results

a) Random Forest: For this model, the best configuration found was:

- Preprocessing
  - trimming using 20 dB below max amplitude;
  - keeping only native American English speakers under the age of 65;
  - no pooling algorithm;
  - time stretching and augmented data;
  - MFCCs, delta and delta-delta features (Mel-frequency spectrogram not included) with

#Samples/FFT	hop length	f_{min}	# MFCC/chunk
2048	512	50 Hz	10

- The final RF model had these hyperparameters:

# features/split	Criterion	Min samples	Bootstrap	Class weight
log_2	Gini	10	False	balanced

- accuracy on *cross-validation set* = 81.2%
- accuracy on *test set* = 90.0%

b) Support Vector Classifier: : The best configuration we have found

- Preprocessing
  - hard trimming at 4 seconds before trimming;
  - trimming using 20 dB below max amplitude;
  - not removing foreigners nor elderly;
  - time stretching and augmented data;
  - no pooling algorithm;
  - not including the first language spoken and current language used for school/work as attributes;
  - MFCC, delta, and delta-delta features with

#Samples/FFT	hop length	f_{min}	# MFCC/chunk
2048	512	50 Hz	10

- The final SVM model had these hyperparameters:

C	kernel	γ	Class weight
2	radial basis	$1/(\#features * X_{var})$	unweighted

- accuracy on *cross-validation set* = 85.2%
- accuracy on *test set* = 95.5%

We have chosen 2 models for the performance evaluation of our project, both have the same SVM characteristics as above, the difference is on after data augmenting to compute the pitch contour either with library-implemented function (submission 15) or by the direct formula (Eq. 1) (submission 17).

IV. Discussion

Test results can be considered satisfactory with the SVM, being relatively close to the accuracy of state-of-the-art algorithms for intent detection [9]. The polynomial and radial basis kernel presented better results than the sigmoid, because the latter acts as an activation function, so the shape of the hyperplane is not as smooth as for the first kernels. The lower values regularization parameter C (0.1 and 0.5) were performing badly in the cross-validation, demonstrating that they were not sufficient to avoid overfitting. With C greater or equal to 1, the results were all above 85% accuracy in the cross-validation phase so they were not high enough to increase the bias to inappropriate levels. As for the regularization factor based on class weights to reduce overfitting of the class with the highest number of samples in the training set performed slightly worse than the unweighted version in both cross-validation and test sets because the distribution of labels in the test set was rather proportional in the one on the training, we can say that with 95% of certainty.

Although having a lower accuracy in the test and validation set when compared to the SVC model, the random forest one also exceeded 90% accuracy. When dealing with the latter model, we noticed that with no bootstrapping on the individual trees of the forest, the accuracy was higher, probably because this data set can be considered to be small, as the amount of samples is not much larger than the number of attributes. Setting the ”class weight” to ”balanced” also improved the accuracy, as the classes were unbalanced in the train set. Also, contrary to our prior belief that by increasing the ”minimum samples split” hyperparameter we would generalize more and get higher accuracies, the optimum value turned out to be 3. Finally, accuracy did not change relevantly by varying the ”Criterion” hyperparameter, so we arbitrarily set it to ”Gini”.

For both models, the pooling algorithm showed not to be beneficial for the accuracy in this case because the attribute reduction did not compensate for the loss of information. For the number of MFCCs, the contrary happened, because by choosing a relatively low number of coefficients (10), the information loss was compensated by the reduction of attributes.

We tried also the K-nearest-neighbour (KNN) model with DTW as a distance metric [10], but the time complexity of our algorithm was too high and we could not run any significant test to evaluate the goodness of the algorithm, although it has been shown in other papers that it can achieve cutting-edge performance [11] [12].

To further improve the obtained results, we could also have tried these promising models, that in other papers were used for speech recognition:

- Hidden Markov models [13]
- Convolutional Neural Networks [14]
- Recurrent Neural Networks [15]

References

[1] A. P. Simpson, “Phonetic differences between male and female speech,” *Language and Linguistics Compass*, vol. 3, pp. 621–640, Mar. 2009.

[2] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. Taylor, “Emotion recognition in human-computer interaction,” *IEEE signal processing magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[3] V. C. Govoreanu and M. Neghină, “Speech emotion recognition method using time-stretching in the preprocessing phase and artificial neural network classifiers,” in *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 69–74, 2020.

[4] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth annual conference of the international speech communication association*, 2015.

[5] K. Kumar, C. Kim, and R. M. Stern, "Delta-spectral cepstral coefficients for robust speech recognition," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4784–4787, 2011.

[6] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning* (E. P. Xing and T. Jebara, eds.), vol. 32 of *Proceedings of Machine Learning Research*, (Bejing, China), pp. 1764–1772, PMLR, 22–24 Jun 2014.

[7] B. A. Sonkamble and D. D. Doye, "An overview of speech recognition system based on the support vector machines," in *2008 International Conference on Computer and Communication Engineering*, pp. 768–771, 2008.

[8] R. Sinha and S. Ghai, "On the use of pitch normalization for improving children's speech recognition," pp. 568–571, 09 2009.

[9] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5754–5758, IEEE, 2018.

[10] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *Proc. VLDB Endow.*, vol. 1, p. 1542–1552, aug 2008.

[11] Y. Permanasari, E. Harahap, and E. Prayoga, "Speech recognition using dynamic time warping (dtw)," *Journal of Physics: Conference Series*, vol. 1366, p. 012091, 11 2019.

[12] B. J. Mohan and R. B. N., "Speech recognition using mfcc and dtw," in *2014 International Conference on Advances in Electrical Engineering (ICAEE)*, pp. 1–4, 2014.

[13] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[14] 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.

[15] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, Ieee, 2013.