NATIVE AND NON-NATIVE ENGLISH SPEAKER CLASSIFICATION

Sanjana Sarda (UCLA), Prathyush Sivakumar (UCLA)

ABSTRACT

Identifying whether a speaker is native or not for a given language can improve the quality of speech recognition for that language. In this project, we attempt to build a system that can identify whether a person is a native English speaker or not with an appropriately high accuracy using an approach involving speech feature extraction and deep learning for speech by recursive neural networks (RNN) based on the LSTM model and FeedForward networks. This is achieved by using two acoustic features: formants and MFCCs. We achieve a test accuracy of 0.846 from the RNN model, which is 30% higher than the test accuracy of 0.545 from our FeedForward network; and a validation accuracy of 0.636 from the RNN which is 13.6% higher than the Feedforward validation accuracy of 0.5.

1. INTRODUCTION

With the increasing popularity of virtual assistants like Google Assistant, Alexa and Siri, voice recognition systems are more important than ever before. There are myriad applications of such a system: forensic analysis, note transcribing etc.

In the context of this paper, our objective is to classify English speakers as native or non-native by using deep learning techniques for speech processing given a speech segment. For the purpose of this project, we use a database of 170 selected audio files from George Mason University which consists of short accented voice clips saying the same paragraph. The final aim of this project is to be able to distinguish different accents/speakers as native or non-native by analyzing the speech features, i.e. formants and MFCCs that we chose to extract from the audio files using speech feature extraction software.

The best accuracy on the test set we achieve is 84.6%. from the RNN model, as compared to the test accuracy of 54.5% from the FeedForward model.

This paper is organized as follows. The approach via choice of features, feature extraction and choice of deep learning models and parameters are described in Section 2. The experiments and the data set used are described and analyzed in Section 3. Finally, conclusions are drawn, and future work is proposed.

2. APPROACH

In this project, the approach involves acoustic feature extraction, feature vector construction, and machine learning techniques. Formants and MFCCs are extracted as features from '.wav' files obtained from the Speech Accent Archive, as previously mentioned. The features are then made into vectors in order to build and train the Feed Forward Network and Recurrent Neural Networks for speech accent classification.

Various MATLAB toolboxes, such as VOICEBOX [2] and VoiceSauce [4], were utilized to extract the desired features. On the other hand, we developed deep learning models to train and test the datasets in Python using Keras, a popular Python deep learning library.

2.1. Feature Extraction

We extracted the Mel-frequency cepstral coefficients (MFCC) from our sample speech samples and fed them as inputs to our RNN model using VoiceBox's [2] y_melcepst. Since the length of each recording was different, the feature extraction resulted in matrices with different sizes for different recordings. In order to feed it to the neural network for processing each of the matrices had to be normalized to the same length. This was achieved by zero padding all the recordings to the length of the maximum recording size.

VoiceSauce was used to extract formants from the voice clips and then feed it to the FeedForward network. Again, since the recordings were of different lengths, we had to zero pad each one to achieve a consistent matrix size.

We chose not to extract certain features if it seemed that they would not contribute to our end goal of classifying English speakers as native or non-native. Efficiency and complexity of our system were major factors too, to not slow down the neural network by processing too many redundant features. We didn't extract pitch as it only gives us information about gender and age and chose not to extract harmonics, amplitude and energy because that helps in identifying different emotions in a speech sample which isn't our goal.

2.2. MFCCS

Our articulations control the shape of the vocal tract. The speech model that we use combines the vibrations produced by the vocal folds with how the vocal tract spectrally shapes the vibrations. The glottal source waveform will be suppressed or amplified at different frequencies by the shape of the vocal tract. Computing the Cepstral will help us to separate the glottal source from the filter.

The log spectrum that the Mel produces is composed of information related to the phoneme and the pitch. The peaks help to identify the formants that distinguish phonemes. We ideally want to separate the pitch components from the phoneme components. As periods in the time or frequency domain are inverted after transformation, we can apply the inverse Fourier Transformation to separate the pitch information from the formants. The log power spectrum is real and symmetric. Its inverse DFT is equivalent to a discrete cosine transformation (DCT).

$$y_t[n] = \sum_{m=0}^{M-1} \log(Y_t[m]) \cos\left(n(m+0.5)\frac{\pi}{M}\right)$$

DCT is an orthogonal transformation. Mathematically, the transformation produces uncorrelated features. Therefore, MFCC features are highly unrelated. In machine learning, this makes our model easier to model and to train. [3]

2.3. Deep Learning Models

For the FeedForward network, the four types of acoustic features were concatenated into a 170 × 4 matrix, which correspond to the means of the first, second, third, and fourth formants extracted from the speech samples. Assume a feature vector f with a sequence length of n, contains four arrays of acoustic features of the same length: f_{F1}, $f_{F2,}$ $f_{F3,}$ and f_{F4}. The feature vector used for training, validation and testing the FeedForward Network is shown below:

$$f^{(i)} = [f_{F1}^{(i)}\ f_{F2}^{(i)}\ f_{F3}^{(i)}\ f_{F4}^{(i)}]$$

For RNN, sequences need to be preserved as it takes them as its input. In this case, the MFCC's extracted from VOICEBOX were used. However, the speech samples used were of different time lengths, which resulted in matrices of varying sizes. Hence, zero-padding was performed to ensure each feature vector was of the same length.

The FeedForward network constructed was a 64-16-2 sequential node model consisting of three Dense layers each separated by two Dropout layers. The first two Dense layers use a ‘sigmoid’ activation function whereas the last output Dense layer uses a ‘softmax’ activation function. After experimentation, we discovered that a rate of 0.1 for each Dropout layer achieved the best overall performance. Additionally, we used the ‘Adadelta’ [5] optimizer as it is an efficient stochastic gradient descent optimization algorithm which requires no manual tuning of a learning rate and appears robust to our deep-learning model.

The RNN is a sequential model comprising of three LSTM layers (64-64-30). The first LSTM is initialized with a dropout rate of 0.1 as it achieved the best overall performance. In addition, a MaxPooling1D layer was added between the first and second LSTM layer with the size of the max pool window equal to 5. We set the strides of the MaxPooling1D layer to 1 as we want it to ideally be as small as possible without being zero. At the very end of the last LSTM layer, a dense output later was implemented with a ‘softmax’ activation. Unlike the FeedForward network, we used the “Adam” optimizer as it was computationally efficient and straightforward to implement.

3. EXPERIMENTS

3.1. Dataset

We used a database of audio files from George Mason University which consists of 170 short, accented voice clips. Out of the 170 voice clips, 70% was used as our training set, 15% was used for testing and 15% was used for validation. From the 170 files, 80 were native speakers from 20 different countries and 90 were native speakers.

3.2. Main Results

When comparing accuracy, the recurrent neural network achieved better results than the feed forward network.

The RNN achieved a test set accuracy of 84.6% with a validation set accuracy of 63.6%, while the FeedForward network achieved a test set accuracy of 54.5% with a validation set accuracy of 50%. The RNN is clearly more accurate, which is probably due to MFCC’s being a more useful tool for classification than the formants. Thus, the MFCC’s are clearly the most important feature in accent recognition, which was expected given their high popularity in other speech recognition machine learning models. However, the results can still be vastly improved, which will be discussed shortly.

4. CONCLUSION

This project seeks to classify English speech passages into native and non-native English speakers. The approach involves acoustic features and deep learning models.

In this project, there are several lessons learned and they are summarized as follow.

A varying data set is useful to prevent the model from overfitting results.

With our limited testing, the RNN LSTM model performs better than the FeedForward model. However, this should be compared using MFCCs for both models with more extensive testing.

The MFCCs have a better performance than the formants. However, this too, should be extensively tested with both RNN LSTM and FeedForward models.

We were able to build a decent Automated Speech Recognition System with the given speech dataset using an RNN based on an LSTM model. For future work, exploring Dynamic Time Warping to normalize all our speech signals might return better results. Also, optimizing the RNN by experimenting with the layers, optimizers and parameters is something we plan on working on.

5. REFERENCES

[1] *Solving the Problem of the Accents for Speech Recognition Systems*. [Online]. Available: http://www.ijsps.com/ uploadfile/2016/0628/20160628103620514.pdf/. [Accessed: 05-Dec-2019].

[2] *Speech Processing Toolbox for MATLAB Mike Brookes, Imperial College*. [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html. [Accessed: 05-Dec-2019].

[3] *Speech Recognition-Feature Extraction MFCC & PLP Jonathan Hui*. [Online]. Available: https://medium.com/@jonathan_hui/speech-recognition-feature-extraction-mfcc-plp-5455f5a69dd9/. [Accessed: 05-Dec-2019].

[4] *VoiceSauce - A program for voice analysis UCLA Phonetics*. [Online]. Available: http://www.phonetics.ucla.edu/voicesauce/. [Accessed: 05-Dec-2019].

[5] *Zeiler and M. D., "ADADELTA: An Adaptive Learning Rate Method," arXiv.org, 22-Dec-2012*. [Online]. https://arxiv.org/abs/1212.5701/. [Accessed: 05-Dec-2019].