Group 29 (S)hawarma (M)anifolds (A)re (I)ntelligent

Ujjwal Tiwari (2018900100), Projit Bandopadhyay, Nitin Raj (201501246), Aniket Bansal (20161152)

smaiStatistical Methods in Artificial Intelligence

Accent Classification:

Audio-Based Language Detection

Project Overview

Accent classification is a challenging natural speech processing task of identifying elements of speech that are likely to encapsulate the accent of non-native language speakers. These computational methods are challenging due to the complex cross disciplinary nature of the problem. Accented speech from non-native english speakers imposes a challenge to speech recognition because accented speech tends to result in phones that are atypical. Speech recognition systems can benefit greatly from accent classification algorithms by switching between speech recognition systems that are trained for particular accent.

In this project, we attempt to solve this problem by classifying 30-second accented voice clips as one of the target classes.

The approach to this task comprises of three major steps:

1. Data collection and preprocessing

2. Extraction of relevant speech features

3. Training a classification algorithm

In this project we experiment with a set of 6 different languages, namely Bengali, Malayali, Telugu, English, Arabic, Mandarin. However, this approach can be extended to a number of different accent classes. We use 2 different types of acoustic features namely Mel Frequency Cepstrum Coefficients (MFCCs) and Chroma Energy Normalized (Chroma CENS). We also apply PCA to capture the principal components of variation of these features. For the purpose of classification experimentation we use a 3-layer convolutional neural network.

Introduction

Speech recognition is an active area of research and has multiple real world applications. One of the major challenges in speech recognition is to understand the speech by non-native speakers. Accented speech tends to result in phones that are not typical of language which makes speech recognition difficult.

Accent detection or classification can improve the quality of speech recognition in that the Automatic Speech Recognition (ASR) system can firstly identify the ethnicity of a speaker and then use the automatic speech recognition system that is trained for the particular accent. In addition to accent recognition system which provides identification of a speaker's ethnicity, is crucial in applications such as crime investigations.

Mathematical Formulation:

For a non-native english speaker *s* whose native language is L_s in the set of all non-English languages *L*, given their *n*-second clip $c_{s,n}$ is the set of all clips *C.* We would like to find a mapping $Q : C \rightarrow L$ such that the occurrence of prediction misses $Q(c_{s,n}) != L_s$ is minimised.

Define the function *F,* for a subset C_n, subset of *C.*

$$f(\Phi, C_n) = \sum_{c_{s,n} \in C_n} \delta(\Phi(c_{s,n}), s)$$

Where $\delta(x, y) = 1$ if $x = y$ or 0 otherwise. The function $f(\Phi, C_n)$ is the number of prediction misses for all $c_{s,n} \in C_n$.

OPTIMIZATION:

Accent classification can be formulated as an optimisation problem. The objective is to find the best mapping Φ^* for the clip set C_n.

In the context of this project, given a 30-second clip $C_{s,30}$ of a non-native english speaker s, we seek to classify the native language L_s of the speaker *s* to one of the six languages: Bengali, Malayali, Telugu, English, Arabic, Mandarin.

Our approach involves feature extraction, feature descriptors and machine learning. Acoustic features like MFCCs are extracted from the .wav files. Principal Component Analysis (PCA) is use for feature descriptors. Convolutional Neural Network (CNN) is used for machine learning.

The best accuracy on on the test set we achieve is 43%.

The subsections are organised as follows. The background and related work is described in the next section followed by Intuition behind using MFCC features and analysis of other speech features used in literature. In the following section, we describe our approach to the problem. The last couple of sections describe experimental results. We also discuss the challenges and methods to further improve our method.

Background and Related Work

There are multiple approaches to the task of accent classification, from the classical Gaussian Mixture Models (GMM) and Hidden Markov Models (HMMs) to machine learning methods of Support Vector Machines (SVMs) and Multi-Layer Perceptron, then further to deep neural learning methods of Long Short Term Memory models (LSTMs) and Convolutional Neural Networks (CNNs).

Accent recognition methods implementing SVMs and HMMs on top of acoustic features have been heavily used and cited in the literature. Only a few studies done on accent detection taking the advantage of advanced models.

As in many other application of neural networks DNN with more than 6-layers and 1024 neural units outperforms shallow neural networks (SNN) with improved accent detection rates across all tested languages.

Accent classification on audio sequence lengths < 30 seconds is more challenging as the number of features in such sequence affect the performance of the classifier inversely. Real life applications such as crime investigation would likely only have limited audio to listen before needing a classification of accent. Additionally in cases of ASR system like a support line, ARS system should quickly be able to recognise accent to be able to switch to

a more appropriate ASR. For usability and response time we believe that the ASR system should be able to classify accents from a voice clip of no more than 20 seconds and identify the ethnicity of the speaker and then switch to an ASR that is trained for the particular accent.

In this work we seek to classify the accents from 30-second voice clips of the non-native english speaker to one of the five languages: Bengali, Malayali, Telugu, Mandarin, English, Arabic.

Sound Perception and MFCC features(Domain Knowledge):

Speech accent recognition is a problem that required domain expertise to pin down the problem perfectly. Understand how we hear sounds and how we perceive leads to better design and implementation of robust and efficient systems for analyzing and representing speech.

The better we understand signal processing in the human auditory system, the better we can (at least in theory) design practical speech processing systems like accent classification, speech coding, speech recognition etc.

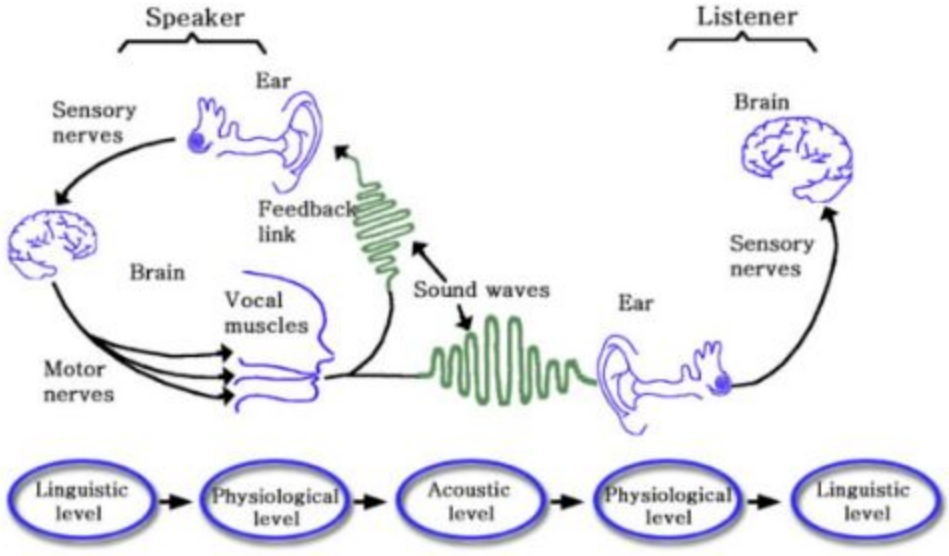

Figure 1. Speech production chain

AUDITORY Models: Most auditory models include the perceptual effects such as

1. Spectral analysis on a **non-linear frequency scale** (usually mel or bark scale).
2. **Spectral amplitude compression (**Dynamic range compression).
3. **Loudness compression** via some logarithmic process.
4. Decreased sensitivity at lower (and higher) based on results from **equal loudness contours.**
5. Utilisation of temporal features based on **long spectral integration** intervals(syllabic rate processing).
6. **Auditory masking** by tones or noise within a critical frequency band of the tone(or noise).

Perceptual effects included in PLP are:

Critical band spectral analysis using bark frequency scale with variable bandwidth trapezoidal filters.

Asymmetric auditory filters with a 25db/Bark slope at the high frequency cutoff and a 10db/Bark slope at the low frequency cutoff.

Use of the equal loudness contours to approximate unequal sensitivity of human hearing to different frequency components of the signal.

Use of the non-linear relationship between sound intensity and perceived loudness using cubic root compression method on the spectral levels.

Other auditory perception models: **Lyon's Cochlear model** and **Ensemble Interval Histogram model.**

MFCC (Mel-frequency cepstral coefficients):

When we hear a sound, it is perceived by our human ear. This perceptual property of the human ear is captured in this technique.

The pair spectrum of the speech signal is converted to the Bark scale which is similar to the human ear's perceptual model.

MFCC is a PLP feature which works in combination of DFT and LP techniques.

Other PLP features include F-Banks, RASTA-PLP etc. In literature we find a heavy use of MFCC, It's deltas, FBanks for automatic speech recognition and accent classification related tasks.

- In MFCC we capture spectral envelope(curve connecting all formants). Perceptual experiments say human ear concentrates on certain regions rather than using whole of the spectral envelope.
- Mel-frequency analysis of speech is based on human perception experiments.
- It is observed that the human ear acts as filter
- It concentrates on only certain frequency components.
- These filters are non uniformly spaced on the frequency axis, more filters in the low frequency regions. Less number of filters in the high frequency region.
- Cepstral coefficients h[K] obtained for the Mel-spectrum are referred to as Mel-frequency cepstral coefficients.

CENS (Chroma Energy Normalized):

To compute CENS features, following steps are taken after obtaining chroma vectors using chroma_cqt:

1. Normalization of the each chroma vector
2. Quantization of amplitude based on "log-like" amplitude thresholds.
3. Smoothing with sliding window.

CENS feature are robust to dynamics, timbre and articulation. Generally used in audio matching and retrieval applications

Process of feature extraction:

- Speech is analyzed over short analysis window
- For each short analysis window a spectrum is obtained using FFT
- Spectrum is passed through Mel-filters to obtain Mel-Spectrum
- Cepstral analysis is performed on the Merl-spectrum to obtain coefficients.
- This speech is represented as a sequence of cepstral vectors
- It is these cepstral vectors which are given to pattern classifiers for the purpose of accent classification.

Approach:

In this project, the approach involves data collection from the speech accent archive from George Mason university. The audio contains structured 30-second speech from 22 languages across the globe. Metadata containing the gender,location, age etc is available with the the audio files. However we do not take use of it. We also collect unstructured from youtube for the task of accent classification in 3 different languages, bengali, telugu, Malayali. Approximately 30 .wav files per class**.**

Data Preprocessing and feature extraction:

The following steps are used for data preprocessing step:

1. Audio o f sequence lengths less than 30 seconds are zero padded to get the sequence length right for every data point.
2. The labels are one-hot encoded into a matrix.(Binary class matrix).

3. Each audio file is searched for silence intervals. These intervals are cropped out of the sequence.
4. MFCC/ CENS feature extraction and Normalisation.
5. For training the CNN on top of this we represent each sequence in a 13*36 array.

Convolutional Neural Network:

CNN applies a series of convolutional filters to raw data or descriptors to learn a hierarchical model of features specific to accent classification task. CNN has three types of layers; Convolutional Layer, Pooling Layer and Fully-Connected layer.

The convolutional layer applies convolutional filters to the input data to capture higher level spatially correlated features. The pooling layers downsample the inputs without affecting the depth. The fully connected layer is the final layer to perform the NN classification.

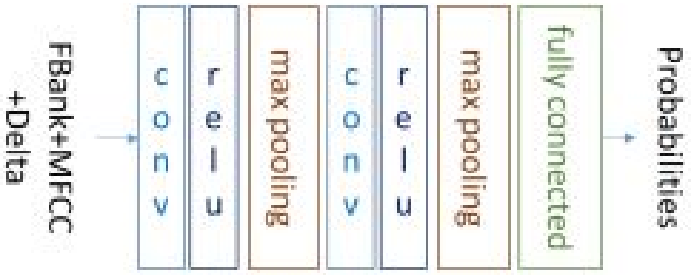

Figure 2: Architecture of CNN.

The above shown is a rough schematic of the CNN. In the convolutional layer we use a convolutional layer of patch (3,3) and a stride of (1,1). In the max pool layer, we use a pooling kernel of (2,2) and a stride of (2,2). This pooling layer does not have overlap in down-sampling. The 'relu' activation function is used for non-linearity. The number of fully-connected unites is tunes.

The network is implemented using KERAS. The maximum gradient norm for gradient clipping is 10, the optimizer is "adam", the exponential decay rate is 0.95.The number of epochs is 100. The dropout rate used in the fully connected layer is 0.1.

Experimental Results:

Test, train split = 0.2

Experiment 1: We train CNN for structured Hindi, Telugu, Malayalam, Bengal audio clips with MFCC features:

Training set includes:

27 samples from hindi, 9 samples from telugu, 16 samples from malayalam , 17 samples from bengali.

Test set includes-

6 samples from hindi, 5 samples from telugu, 4 samples from malayalam , 3 samples from bengali.

Confusion Matrix:

3	0	0	0
0	0	5	0
0	0	6	0
2	0	1	1

Accuracy: 0.55

Experiment 2: We train CNN for structured Hindi, Telugu, Malayalam, Bengal audio clips with Chroma_cens features:

Training set includes:

27 samples from hindi, 9 samples from telugu, 16 samples from malayalam , 17 samples from bengali.

Test set includes:

6 samples from hindi, 5 samples from telugu, 4 samples from malayalam , 3 samples from bengali.

Confusion Matrix:

2	0	1	0
0	0	5	0
1	1	4	0
2	0	1	1

The accuracy for this experiment is 0.37

Experiment 3: We train CNN for unstructured Hindi, Telugu, Malayalam, Bengal audio clips extracted from youtube with MFCC features: (with same train-test split):

Confusion Matrix:

2	0	1	0
0	0	5	0
0	0	6	0
2	0	1	1

The accuracy for this experiment is 0.42

Experiment 4: We train CNN for unstructured Hindi, Telugu, Malayalam, Bengal audio clips extracted from youtube with chroma_CENS features: (with same train-test split):

Confusion Matrix:

2	0	1	0
1	0	3	1
1	1	4	0
2	0	1	1

The accuracy for this experiment is 0.27

As the training data is less we also experimented with other languages that had more data available for training:

Experiment 5:

.Test, train split = 0.2

Experiment 1: We train CNN for structured Mandarin, Arabic, English, Hindi audio clips with MFCC features:

Training set includes:

160 samples from English, 100 samples from Mandarin, 66 samples from Arabic, 33 samples from Hindi.

Accuracy: 0.75

Remark:

We notice that the accuracy changes with the random train test split.

This is due to a lot of gender and other bias in the data collection method

Qualitative Analysis and future work:

1. We need a lot more data to leverage the benefits of deep learning methods like CNNs, LSTMs
2. We would have liked to used some sequence modelling technique like HMMs and LSTMs and extend our model for other features like FBANKS.
3. The performance of CNN benefitted from the large amount of data per class for the last experiment. However, we wanted to experiment with indian languages. Building a standard speech accent corpus for Indian accents can greatly benefit pattern recognition algorithms to perform better.

Dependencies:

- **Python 3.x**
- **Keras**
- **Numpy**
- **Beautiful Soup**
- **Pydub**
- **Scikit-learn**
- **Librosa**

Other sources of data available can be found on the following link:

1. CSLU 1.2 [link]
2. CSLU 22 [link]
3. The speech accent archive. [link]
4. Non-native speech database. Link
5. Wild-cat corpus of Native and Foreign accented english. [link]

[GITHUB_LINK]

Work Distribution

- Ujjwal: Model Creation and testing
- Projit: Model Creation, data collection
- Nitin: Feature selection, data collection
- Aniket: Experimentation, data collection