Automatic Speech Recognition - Design and Implementation

Zhongye, Student ID. 100334296

December 23, 2020

Abstract

Speech is a form of communication technique between humans, but now a day's humans and machines are communicating with each other using various techniques like speech recognition, speech enhancement and speech synthesis. Over years of research and development in this field, the most challenging part is the accuracy of the speech recogniser. This paper attempts to discuss in detail about the design and implementation of an automatic speech recogniser using Matlab, SFS and HTK with high accuracy. Speech recognition system focusses on speech collection, normalization, labelling, feature extraction, acoustic modelling, language modelling and noise compensation.

Keywords: Automatic speech recognition, Acoustic model, Language Model, Speech Processing, Pattern Recognition, Hidden Markov Model.

1 Introduction

Automatic speech recognition (ASR) is the process of converting a recorded speech signal into text. Speech processing is one of the most exiting areas of research for the decades, ASR made the computers to understand human languages and follow the instructions. Due to the advances in statistical modelling, ASR finds wide range of applications in the tasks that require human machine interface like automatic call processing, helping differently abled people, online banking using voice, transcribing, taking satisfactory survey, virtual digital assistants like siri, alexa, cortona, . . .

The challenging part of any ASR is its accuracy, which depends on various factors like speaker, vocabulary, environment, acoustic modelling, language modelling, filterbanks, transducer, incompatibility between train and test datasets . . . This paper focusses on various methods to improve the accuracy of the ASR. [1][2][3][4]

2 Data collection

Collecting speech samples is the first and foremost step while building a speech recogniser, the samples collected will be used for training and testing the ASR. Speech can be recorded either in Matlab or SFS with suitable sampling frequency. Sampling frequency or the sampling rate is the number of samples per second. Leading zeros should be removed while saving the speech signal. When the signal amplitudes exceed 1, the speech signal will be clipped, to avoid clipping the signal has to be normalized. Figure 1 shows how the recorded signal should be.

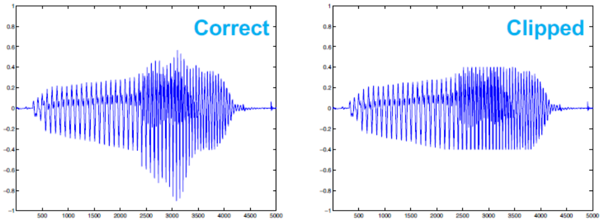


Figure 1: Sample speech signal

If the speech signals are collected with 2 different sampling frequencies, to maintain uniformity, they have to be resampled to a common sampling frequency using resample function available in Matlab.

2.1 Annotation

Annotating or labelling the recorded signal can be done using SFS (Speech Filing System), SFS generates the narrowband and wideband spectrograms, which will help in deciding the voiced content, unvoiced content and silence in the signal. This can be labelled and saved to .lab files.

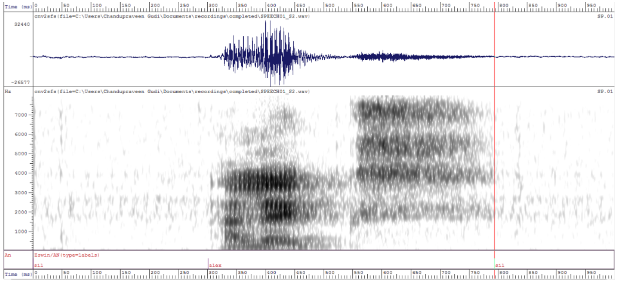


Figure 2: Labelling a speech signal

3 Feature extraction

For a speech signal to be processed, it must in a quasi-stationary state. The speech signal must be divided into small blocks so that the signal is almost stationary and can be processed easily. The size of the small block can be chosen as 20msec as the human articulators cannot move faster than 20msec.

3.1 Pre-emphasis filter

Higher frequencies need to be emphasized as LPC (Linear Predictive Coding) equations tend to satisfy more on lower frequencies leaving behind the higher frequencies while performing LPC analysis. Energy of low frequency sounds is higher than the energy of high frequency sounds due to the radiation properties of the mouth. We apply a pre-emphasis filter to spectrally flatten the energy at all frequencies i.e. making the energy levels equal at all the frequencies. This can be done using filter function in Matlab and the output will be as shown in Figure 3.

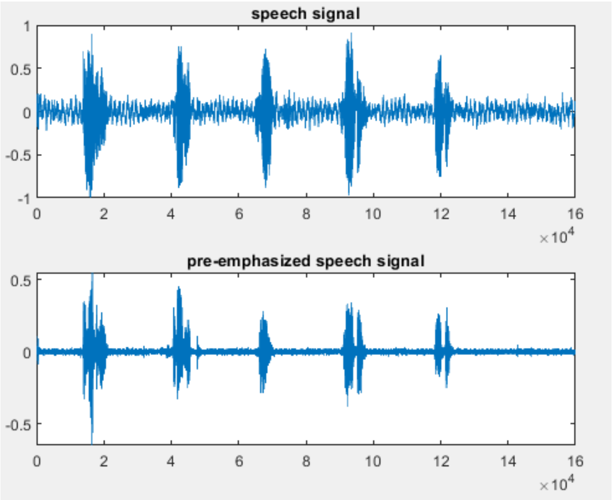


Figure 3: Pre-emphasized speech signal

3.2 Windowing and DFT

Windowing techniques are often used to improve the spectral characteristics, Hamming window is one among the windowing techniques available, shape of the hamming window is designed to minimise the spectral distortions, applying a hamming window with an overlap of 50 percent to the speech signal will reduce the spectral leakage. Size of the hamming window should be the same as size of a small block of speech signal. A predefined function named hamming is available in Matlab for applying it to the speech signal. DFT is often used to convert the signal from time-domain to frequency-domain. FFT function in Matlab can be used to transform the signal, there by magnitude and phase spectra can be calculated. Phase spectrum can be ignored as the human ear is not sensible enough to perceive the changes in phase.

3.3 Filterbank

Filter bank is used to quantize the spectrum across frequency, which will produce the spectral information as per the number of channels selected. For eg. if the spectrum has 160 samples and with 20 filterbank channels selected, the output will be 20 filterbank energy vectors with 30 coefficients in each vector. Melscale filterbank is the most commonly used filterbank, this is a series of triangular bandpass filters, which mimics the human auditory system. The filterbank is based on a non-linear frequency scale called the Mel Scale. Filters are overlapped such that lower boundary of one filter is situated at the center frequency of the previous filter and the upper boundary is situated at the centre frequency of the next filter as shown in Figure 4.

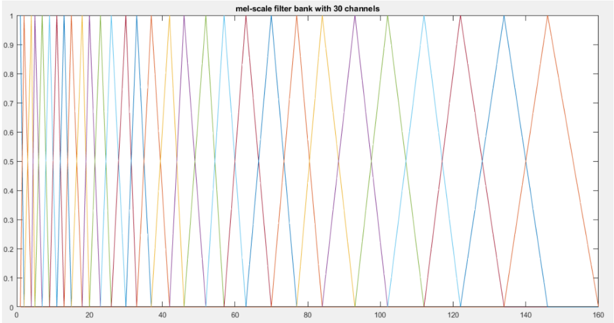


Figure 4: Melscale filterbank

3.4 Logarithmic of filterbank energies and DCT

The logarithmic of filterbank energies separates gain, vocal tract and log excitation signal. Usually mel-frequency cepstral coefficients or mel-cepstrum can be computed by applying the Discrete Cosine transform to the logarithmic of filterbank energies. With the MFCC coefficients, vocal tract information and pitch information can be separated. Pitch information can be ignored as it is not important for the english language.

3.5 Energy component

Energy component can be used as an additional element in the MFCC vector to boost the accuracy of the model.

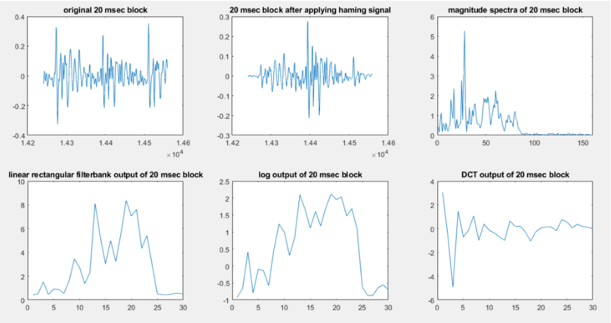


Figure 5: Phases in feature extraction

4 Acoustic Modelling

An acoustic model is used in speech recognition. The model is learned from set of audio recordings and their labelling files to each word.

4.1 Data preparation

4.1.1 Dictionary File

Prepare vocabulary list including all words of the speech file.

4.1.2 GRAM File

The grammar file first contains all the words that need to be recognized. Then the grammar file needs to formulate grammar rules or form a connected vocabulary string. In particular, the sil after the word can be omitted, which may improve the accuracy a little.

4.1.3 NET File

Use HParse command in HTK to generate network file based on grammar file.

4.1.4 HMM Prototype File

The prototype file needs to determine the number of states. The number of Gaussians per states and the kind of parameter are determined when the MFCC file is generated.

4.1.5 Labelled File

The list file is generated according to the path of the MFCC file.

4.2 Training the model

The HInit command can train the model of each word separately based on labelled file, MFCC file and HMM prototype file. The HRest command applies Baum-Welch to retrain a new model based on HInit's HMM model.

4.3 Testing the model

The HVite command to test the accuracy of the modals based on the grammar and dictionary files prepared in advance. Then HResult command use the confusion matrix to display a more intuitive accuracy rate.

4.4 Noise compensation

When the model is trained without noise, the model performance is very poor when tested in noise environment. Model is retrained with some noise signals and tested, the accuracy of speech recognition get better.

5 Testing and Evaluation

Two testers A and B, 20 names and each name 20 times, totally 400 speech signals. Everyone uses half of the 400 speech signals to train acoustic model, and then the other tests the accuracy of speech recognition. The difference is that tester A is from India and uses 16000hz and 16000 samples per second to collect 400 speech files. Tester B is from China and uses 40000hz and 40000 samples per second to collect 400 speech files. They are all recording speech in very quiet conditions. The other conditions and parameters are the same, 8 states, 20ms for frame period, 10ms for hamming window, 31channal including 30 vector and one energy component.

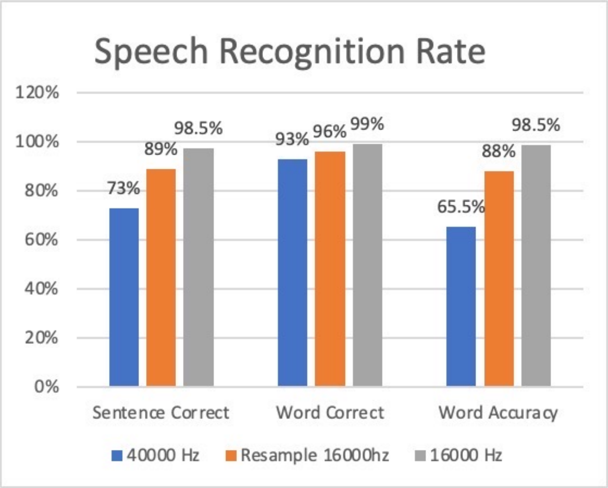
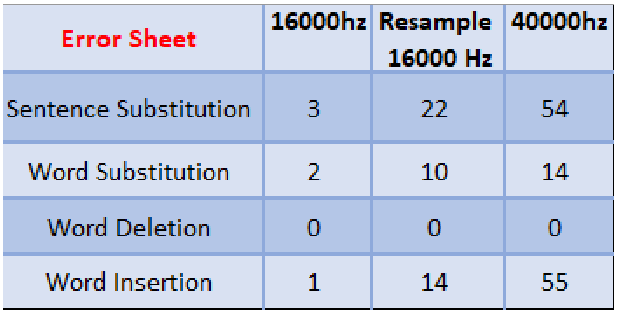


Figure 6: Accuracy Rate



Error Sheet	16000hz	Resample 16000 Hz	40000hz
Sentence Substitution	3	22	54
Word Substitution	2	10	14
Word Deletion	0	0	0
Word Insertion	1	14	55

Figure 7: Error Table

Experimental results show that the accuracy of 16000 samples per second is greatly improved regardless of the type of sentence error or words errors. Each speech file only contains one English word. The meaning of the sentence is silence-name-silence in this experiment. Both of them have no deletion error when recognizing words, probably because the time of each test speech file is very short, and the collection environment is very quiet. In the experiment of 40,000 samples per second, sentence substitution and word insertion are both very high. The reason may be that the data collected is too detailed so that human ears can't distinguish the difference, but the machine can be distinguished. The conclusion is that there is no positive correlation to collect more detailed data and the performance of acoustic model. However, considering that there may be problems such as different testers' collection methods, the way of labelling files, even different countries and accent, the experimental results may not accurate. Therefore, using the resample function of MATLAB adjust the speech file of tester B from 40000 samples per second to 16000 samples per second. Although after resampling, tester B correct rate is still not as good as tester A. But if only compare the 40000hz and 16000hz of student B, 16000 still has better performance. Especially the insertion errors of words and the substitution error of sentence are greatly reduced.

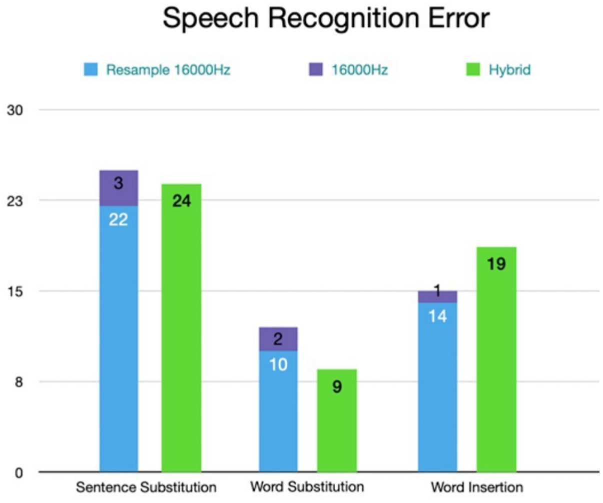


Figure 8: Speech Recognition Error

In addition, a new experiment simply called Hybrid experiment that total of 800 voice files were used to come from two testers, 400 files resampling to 16000hz to 400 files of tester B and A. The new results show that they are basically the same as the sum of the A and B experiments. The reason may be that the number of our collection is still too small. It may also be that the impact of two different countries' accent is not great.

The previous experiments are based on the same number of states, same frame period in each frame, the same number of channels and the same type of filterbank. Later experiments will explore the impact of these four variables on the speech recognition. The new experiment will first be divided into two parts, one part analyzes word errors in the speech recognition, and the other part analyzes some errors occurred in recognizing complete sentences. Simultaneously, the test discusses four variables to affect the results of speech recognition, the time of each frame, the number of channels, the kind of filterbank. The control group is in four states, four milliseconds each frame or 320 samples each frame, thirty channels of each vector, linear rectangular filterbank. The treatment group will increase and decrease the frame period to 2ms and 8ms respectively, Increase the number of States to 8 States and decrease to 2 States, increase the number of channels to 60 and decrease to 15 channels.

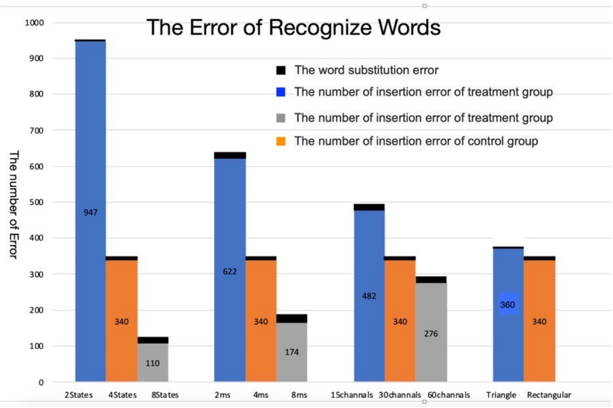


Figure 9: Word Error in Speech Recognition

It can clearly see that the substitution error only occupies a small proportion in the process of speech recognition. The number of states increases, the number of insertion errors will decrease. On the other hand, although Increasing the number of states will lead to a higher likelihood. Unfortunately, increasing the number of states will also lead to a huge increase in the number of parameters (transition probabilities, initial state probabilities). In the term of the second variable, the increase in the period of each frame also means that the number of vectors in each MFCC file decreases, or that using less data to train the acoustic model. In other words, collecting more data does not necessarily lead to better accuracy. But it is obvious that if divide each second into 5 frames or even one frame, the accuracy rate will definitely drop greatly. Therefore, the appropriate frame period is very important, not necessarily high or low, it should be based on different situations. In addition, the increase in the number of channels can also reduce the number of errors and increase the accuracy. This also means more data is extracted in each frame in the process of feature extraction. In particular, what is the difference from the previous experiment is that this parameter is to improve the accuracy of each frame, not the number of frames. The last variable is a different kind of filterbank. Theoretically, it should be that the triangular filterbank has better performance Because the triangle filterbank can imitate the channel of frequency that the human ear can distinguish to a greater extent. It may be that there are too few speech files in our test, and it is difficult to get any effective results.

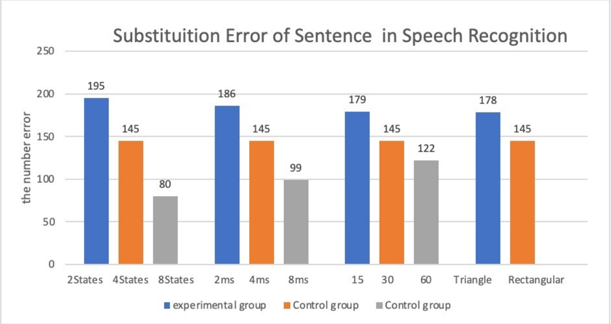


Figure 10: Sentence Error in Speech Recognition

Although there is only substitution Error in sentence recognition, the overall trend is the same as recognizing words.

6 Discussion

Due to the test files and training files in this experiment are relatively small, so choosing such a poorly accurate control group. Changing variables can more intuitively feel the difference in results. How to find the appropriate frame period and appropriate the number of samples per second is very important. According to the current results, the increase in the amount of states brushing will improve the accuracy, but the best performance of what number of states is important. This should be based on different scenarios, there will be a best value. Although using machine learning may find a formula that can adapt to most scenarios, maybe the best speech recognition should be based on different scenarios to train a special speech recognizer. On the other hand, speech enhancement and noise removal are also very important. It may be possible to collect noise in advance a few seconds before turning on speech recognition and perform speech enhancement based on the collected noise to improve accuracy.

References

[1] M.A.Anusuya, S. (2009). Speech recognition by machine: A review. *International Journal of Computer Science and Information Security,*.

[2] Mcloughlin, I. (2009). *Applied Speech and Audio Processing*. Cambridge University Press.

[3] Milner, B. (2020). Audio visual processing lecture notes. *University of East Anglia*.

[4] Shipra J. Arora, H. R. P. S. (2012). Automatic speech recognition: A review. *International Journal of Computer Applications (0975 – 8887)*, pages 0975–8887.