Speech recognition using Dynamic Time Warping and MFCC

Gonzalo Ruiz, Rafael A. Cisneros

***Abstract*—the aim of this paper is to explain the process of speech processing. It will start contextualizing the problem and analyzing the difficulties in the attempts to solve the problem, describe the process, and end with a technical solution that can be applied in isolated word processing.**

Due to the wide variety of speech recognition systems, the system analyzed in this paper will be narrowed down to an isolated word processing, which means each word is spoken within a separated time frame, and there is the consideration that the environment provides little to no noise to the signal. Also, the algorithm will not discern grammatic accuracy, since this is not a capability supported by isolated word recognition systems.

***Index Terms*—Speech Recognition, Dynamic time Warping, Mel Coefficients, MFCC.**

I. Introduction

AN ideal speech recognition system is not invented yet. There are a vast number of problems that affect the whole process and the complexity of it. The application hereby presented will be speech recognition in a quiet environment and using isolated words. In order to recognize a word, the system must first be trained. The way this is going to work is by comparing a received signal with a database of known signals previously recorded. That said, there are two phases: training and application. The training phase consists on recording the different type of signals the system will later recognize. To make this accordingly to the characteristics of the system previously mentioned, the signal must be an isolated word, recorded in a quiet environment. The storage of these signals is done by saving a vector of coefficients known as the Mel coefficients, which characterize the signal. According to the Institute of Space and Technology in Islamabad, this method has an estimated accuracy of 90%.

II. Process

The algorithm consists on processing the signal by finding its Mel Frequency Cepstral coefficients and then by comparing them with the ones of the recorded signals that we are using as reference. The steps are described as follows:

A. Detection

First, **the signal is recorded**, and it is sampled with an average duration of 3 seconds, the average duration of words, at 8000 samples per second, which is above the Nyquist frequency for the frequency range of human audition, which ranges to up to 3400Hz.

Afterwards, the **signal must be partitioned in the area of interest**, which is where the word information is contained. This simple step will reduce the samples to analyze from 24000 to approximately 5000. This will improve the efficiency and accuracy of the algorithm.

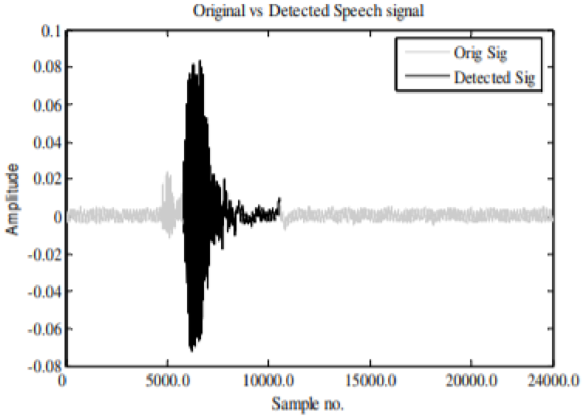


Figure 1: Example of the original vs detected speech signal when the speaker says "one" to the microphone .Reference: [6]

B. Pre-Emphasis

In this phase, an algorithm **increases the energy of the signal proportional to its frequency in every point**. By doing this, a more uniform signal is obtained, which can be analyzed better. For that, we apply what is called a "pre-emphasis filter", which can be expressed with the following formula:

$$H(z) = 1 - \tilde{a} * z^{-1} \qquad 0.9 < \tilde{a} < 1.0$$

Typically, the value used for ã is 0.95. With this step the noise of the signal is minimized by giving energy at high frequencies, which are associated to voice (environmental noise is associated to low frequencies).

C. Frame-Blocking

Thirdly, the signal enters a frame blocking phase, in which **the signal is separated into frames of N samples**, each separated with each other by M (N>M). If N is increased, frequency resolution is increased. On the other hand, the lower the N, the

April 28th, 2019. TECNUN, Universidad de Navarra. Señales y Sistemas.

more local spectral properties that can be analyzed, so there must be a balance, depending on whether the objective is to have higher frequency resolution or analyze local spectral properties. Normal values for N and M are 256 (~30ms) and 100, which are the ones to be used, but values of 20 ms are also typical. This process makes the signal a discrete one.

D. *Windowing*

After being frame-blocked and made a discrete signal, the signal must be windowed. This translates into a **smoothing and minimization of discontinuities at the edges of each frame**, which in this case is done by multiplying the values of the signal by a spectral window, which in this case is the Hamming window, as described below:

Hamming window:

$$w(k) = 0.54 - 0.46 \cdot cos\left(\frac{2\pi k}{N-1}\right)$$

When 0 <= n <= N-1.

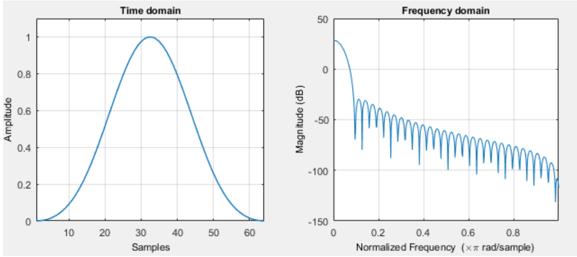


Figure 2: Graphic of the Hamming window both in the time and the frequency domain.

On the time domain, applying this window will minimize the edges of the signal. In this specific case, it is necessary to apply the Hamming window in order to minimize the discontinuities formed at the edges, which means that we must apply it so that the discontinuities near the edges (near the 0 and N samples) of each partition of the signal are minimized.

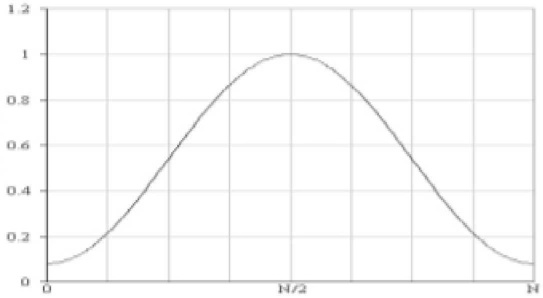


Figure 3: Graphic of the Hamming window in the time domain in the case considered. Reference: [6]

E. *Fast Fourier transform*

The Fast Fourier Transform (FFT) is an algorithm which applies **the Fourier transform to the frame-blocked discrete signal in order to have its components in the frequency domain**, in which the noise can be minimized. This is because "*different timbres in speech signals correspond to different energy distribution over frequencies*[1]".

Then, with the signal on the frequency domain, the energy distribution can be found. The Fourier transform is defined as the following:

$$S_n = \sum_{k=0}^{N-1} s^{-2\pi jkn}/_{N}$$

Where S_n is the FFT of one windowed speech frame, with coefficients being S_k.

F. *Mel Filter banks*

The signal is now a discrete signal in the frequency domain, where the Mel filter banks are applied. **The Mel Filter Banks are band pass filters** which *"Mimic the auditory system, so it is a perceptual scale of pitches based on the known variation of the human ear's critical bandwidths*" [3].**This filter captures the phonetically important characteristics of speech**.

The Mel scale represents linear frequency spacing below 1000 Hz and a logarithmic one above 1000 Hz. In order to convert the ordinary frequency scale to the Mel scale, in which subjective pitches of the human voice can be measured, the following equation is applied:

$$Mel(f) = 2595 \cdot log_{10}\left(1 + \frac{f}{700}\right)$$

The Mel frequency graph is plotted by overlapping the lower boundary of one filter at the center frequency of the previous filter and the upper boundary is situated at the center frequency of the next filter. In the end the result will be:

$$X_m = \sum_{n=1}^{\frac{N}{2}-1} |S[n]| \cdot |H_m[n]| \quad 1 \leq m \leq N$$

In which:

X_m is the filter output
$|H_m[n]|$ is the frequency magnitude
S[n] is the N-point discrete frequency-domain signal, obtained in the Fast Fourier Transform process.

Which leaves a Mel frequency graph like the following:

[1] According to the International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, expressed on the Vol. 2, Issue 8, August 2013, in reference to the application of Fast Fourier transform algorithms to Speech recognition and analysis.

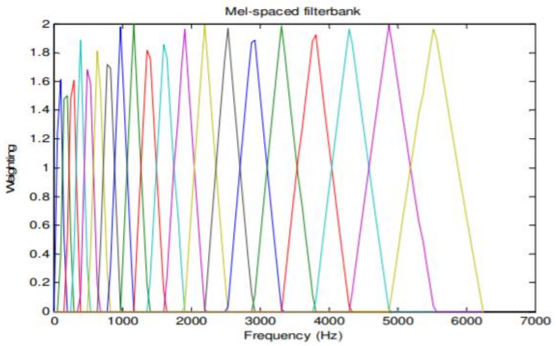


Figure 4: Graphic of the Mel Filter Banks overlapped one into the midpoint of the previous one. Reference: [6]

This graph has this form because it is the consequence of overlapping the successive Mel Filter Banks as described above.

G. *Logarithmic compression*

Afterwards, a quick logarithmic **compression is applied to model the perceived loudness of a given signal intensity**. This is done by applying the following expression:

$$X_m(ln) = ln(X_m) \quad 1 \le m \le N$$

Discrete Cosine transforms
Finally, a Discrete Cosine Transform is applied to **convert the signal from a logarithmic Mel signal back to the time domain**. The result of this transform is what is called the Mel Frequency Cepstral coefficients. This is the equivalent of the Real part of the Fourier transform of a signal:

$$MFCC_k = \sqrt{\frac{2}{M}} \sum_{m \equiv 1}^{M} X_{m(ln)} \cdot cos\left(\frac{\pi k(m - 0.5)}{M}\right) \quad 1 \le k \le p$$

These coefficients (12~20) are stored in memory and represent a word.

H. *Comparison*

The comparison between the coefficients and the database signals is the Euclidean distance between them:

$$d(x, y) = \sqrt{(x - y)^2}$$

Even though this looks easy, it is not computationally cheap for a couple of reasons:

First, all the comparisons are to be done between the reference utterance and the input that was captured with the microphone. This means that, before applying the process described above, a database must be created with all the possible references which are the ones that are going to be compared with the input. This database is also known as the dictionary of the algorithm. It is important to remark that cost (understood as computation complexity) increases exponentially with the number of words stored in the dictionary, so there is a need to balance the number of words with the efficiency of the system.

Each one of the elements on the dictionary must be recorded previously. But since the comparison method is based on Mel's Cepstral coefficients, the continuous signal is not needed. It's enough to apply the process above and store Mel's coefficients for each one of them. That will leave the dictionary with only numeric vectors on it (usually with a length of 12 ~ 20 numbers per signal), which is much more convenient for analysis.

So, in theory, the comparison will be the Euclidean distance between each of the components of the vectors between them, described by the following equation:

$$d(x, y) = \Sigma\sqrt{(x_i - y_j)^2}$$

Where (i,j) go from 0 to the length of the vectors, called N:
$0 \le \mathrm{i} \le \mathrm{N} \quad 0 \le \mathrm{j} \le \mathrm{N}$

But there is a problem, if these vector/strings are not the same length (which is usually the case), the distance cannot be properly calculated, and a technique called Dynamic time warping must be used.

I. *Dynamic Time Warping*

DTW is an algorithm used to compare two signals of different duration. This is done by matching appropriate regions of the test utterance with the appropriate regions of the reference utterance. In other words, align the reference and test features.

To accomplish this, a grid is used. This grid is divided into squares, and each of them represents a letter. In the vertical axis the reference utterance and on the horizontal axis, the test utterance. This means that the problem can be solved by dividing the word into letters and comparing them one by one. For that, we must draw a path that goes from the first letter of the test utterance to the last one according to the following rules:
- The only possible directions are: →, ↗ and ↑.
- If it goes from one letter of the test utterance to the same one, the direction is →.
- If there is a jump from a letter to a different one in the test utterance and it matches the one of the reference utterance, the direction is ↗.
- If there is a jump from a letter to a different one in the test utterance, but this one doesn't match the one of the reference utterance, the direction is ↑.

This way, a path is created that represents the minimum distance between the words, which is the sum of the individual distances between letters. For example:

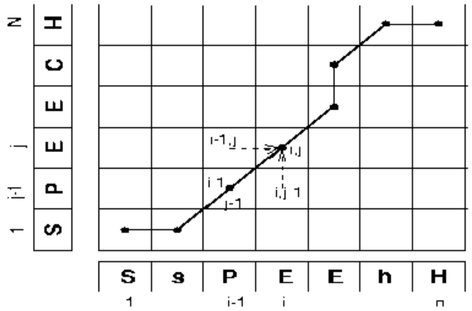


Figure 5: Example of the Dynamic Time Warping grid with the same word. Reference: [6]

The path describe is at follows: First, it goes → because it compares the same letter (S - s). Then, ↗because it goes from one letter of the test utterance to a different one (S-P) and it matches the one of the reference utterances. Then, the path goes through 2 different letters in a row (P-E, E-E) of the test utterance that match the reference, so that's ↗ twice. Then, it goes upwards because it would go from an E to H, which doesn't match the next letter of the reference, which is C. Then it goes ↗ because the letter changes in the test utterance (E-H) and it matches the H on the reference and, finally,→ because the comparison goes through the same letters (H-H). That path is the optimal one.

The only thing left is calculating the distance between each letter and then do the sum. Note that each direction has different distances associated: → will be the one with less distance (since letters are the same) and ↑is the one with the most distance (since letters are completely different). Therefore, comparing words with different letters in the reference and test utterance will give a lot of ↑ directions, thus giving a lot of distance, which accomplishes the objective. Another example of grid would be:

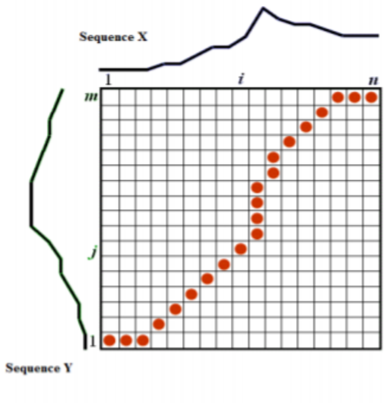
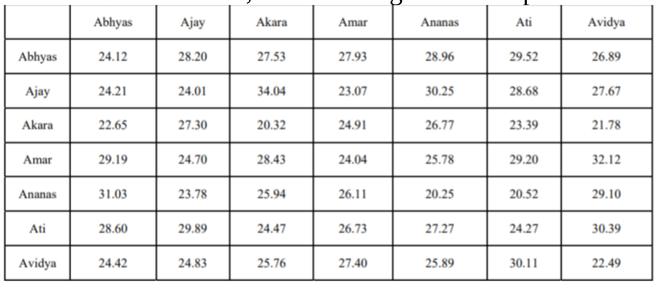


Figure 6: Example of a generic Dynamic Time Warping grid. Reference: [8]

This process is repeated with the reference utterance and each word on the dictionary, comparing the input word with all the rest. The one with the minimum distance calculated will be the word. The total distances can be represented by a table of distances. For instance, the following is an example of one:



	Abhyas	Ajay	Akara	Amar	Ananas	Ati	Avidya
Abhyas	24.12	28.20	27.53	27.93	28.96	29.52	26.89
Ajay	24.21	24.01	34.04	23.07	30.25	28.68	27.67
Akara	22.65	27.30	20.32	24.91	26.77	23.39	21.78
Amar	29.19	24.70	28.43	24.04	25.78	29.20	32.12
Ananas	31.03	23.78	25.94	26.11	20.25	20.52	29.10
Ati	28.60	29.89	24.47	26.73	27.27	24.27	30.39
Avidya	24.42	24.83	25.76	27.40	25.89	30.11	22.49

Figure 7: Example of a table of distances calculated with the Dynamic Time Warping method. Reference: [7]

Note that the minimum distance is the one where the reference and test words are the same

III. Conclusion

Knowing this algorithm is a limited version of the overall speech recognition problem and taking into consideration everything mentioned above, problems arise if there were to be a high amount of words, comparing the distance between all words and the reference, not only because of computational cost, but the chances of an error are high, since it would be easy for the program to get confused with words like "weak" and "week", which are phonetically similar. That is the main reason behind the MFCC method using a small dictionary, and the main function of it to be to recognize simple commands such as "open", "close", "yes", "no", etc.…

Therefore, although it's a limited algorithm, it's used in almost every feature of speech recognition, either as the core of the program or as a tool for more complete algorithms.

IV. Bibliography

[1] John R, Deller, "The speech recognition problem" in *Discrete-Time Processing of speech Signals, published on*IEEE press, October 1999

[2] John R, Deller, "Dynamic Time Warping" in *Discrete-Time Processing of speech Signals, published on* IEEE press, October 1999

[3] Anjali Bala, "Voice command recognition system based on MFCC and DTW," *International journal of Engineering Science and Technology*, vol. *2*, no. *12*, 7335-7342, 2010.

[4] Lindasalwa Muda, "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) techniques", *Journal of Computing*, vol. 2, issue *3*, 2151-9617, 2010.

[5] Lawrence R. Rabiner, "Considerations in Dynamic Time Warping algorithms for Discrete Word Recognition"*IEEE Transactions on acoustics, speech and signal processing, Vol. ASSP-26, no. 6, December 1978.*

[6] Talal Bin Amin. (2008, November). Speech Recognition using Dynamic Time Warping. Presented at Conference "2nd international Conference on Advances in Space and Technologies. [ICAST]. Available: online.

[7] Shivanker Dev Dhingra, "Isolated Speech Recognition using MFCC and DTW" Internationa*l journal of Advanced Research in Electrical, Electronics and Instrumentation Engineer* vol. *2*, no. *8*, 2278-8875, August 2013.

[8] Bharti W. Gawali. (2010) Marathi Isolated Word Recognition System using MFCC and DTW Features. Presented at Conference "Proc. of international Conference on Advances in Computer Science". Available: online.