CONTINUOUS FORMAL ARABIC SPEECH RECOGNITION SYSTEM BASED ON HIDDEN MARKOV MODEL

Sa’ed Abed
Department of Computer Engineering, Kuwait University, Kuwait

Hanem Ellethy
Department of Computer Engineering, Kuwait University, Kuwait

Mohammad AlShayeji
Department of Computer Engineering, Kuwait University, Kuwait

Abstract—Speech recognition is a mechanism to recognize words and phrases in any language and translate them to a machine-readable layout. Recently, Automatic Speech Recognition (ASR) systems are used widely in many applications like translating speech to text, home security systems, and military applications. Unfortunately, the speech recognition field for the Arabic language is not yet mature and it needs more development. The main objective of this paper is to implement and evaluate a continuous speech recognition system for the formal Arabic language using Mel-Frequency Cepstral Coefficients (MFCCs) and Hidden Markov Model (HMM) with Gaussian Mixture Model (GMM). Adding the energy feature to the MFCCs twelve features for each frame has a good impact on the accuracy for the continuous speech. In addition, using GMM was effective as a post processing step before calculating the HMM parameters for training. A recognition rate is enhanced to be 94% as a result of tuning parameters and using different techniques. The Arabic speech recognition system is built using MATLAB based on HMM toolbox. A comparison between our work and the state of the art works is presented to show the acceptability and intelligence (accuracy) of our work.

Keywords—Arabic Speech Recognition, Artificial Intelligence, GMM, HMM, MFCCs.

I. INTRODUCTION

The speech is considered the most important means of human's communication. Automatic Speech Recognition (ASR) systems are used to convert the speech to its equivalent text using computer programs [1]. Nowadays, ASR is a significant key in human-machine communication especially after the enormous development of mobile-based systems and applications. This technology uses the spoken input instead of typing, clicking or selecting to trigger some action to increase productivity and make software more user-friendly. Over the last decade, speech recognition technology has begun to appear in many voice applications including phone dialing and routing, data entry, translating speech to text, and in a multitude of devices from home computers to mobile phones, home security systems and, military applications.
ASR has numerous applications that are used in many sectors. In education area, it is used in vocabulary pronunciation correction as a help in foreign languages learning and many other applications. Moreover, ASR has diverse utilization in computer games, robotics, domestic appliances, healthcare, commercial domain, and military sectors as in high performance fighter aircraft [2].

Arabic spoken language is one of the most broadly used languages of the world. Arabic speech recognition systems are limited compared to other languages because of many dialects of Arabs and morphological complexity [3]-[5].

Building an ASR system passes through some steps to achieve its target. First, preprocessing the audio signals (speech) then extracts its features, and the last step is using recognition system to realize and identify these features.

A typical architecture of ASR system is illustrated in Fig. 1. The signal processing and feature extraction takes the input audio signal and applying noise filtration, then converts the signal from time to frequency-domain and extracts the feature vectors. The Acoustic Model (AM) takes input the feature generated from the feature extraction component, and generates AM score. After that, the Language Model (LM) estimates the probability of proposed word sequence by learning the relation between words from the training corpora [6].

![Fig. 1 Architecture of ASR system](image)

Human speech has different shapes depending on the vocal cords, tongue, lips, jaw, and others. Mel-Frequency Cepstral Coefficients (MFCCs) are good and accurate techniques that can represent the speech shape in the form of short time power spectrum and extract its features [2], [7], [8].

On the other hand, Hidden Markov Model (HMM) is a statistical model that based on Markov process with invisible states and visible output tokens. The probability distribution of each state is formulated based on the possible output tokens. The series of tokens point to some information about the series of the hidden states [3]. HMMs are immensely used in many ASR systems for different languages [9]-[13]. This is because HMMs are trained automatically, simple, and have an effective computations.

The aim of this paper is to present an overview of speech recognition technology, definitions, algorithms and tools concentrating on the Arabic speech recognition system. The speech recognition field for the Arabic language is not yet mature and it needs more development. A continuous formal Arabic Speech Recognition system using MFCCs and HMM is implemented and evaluated. The system is built using MATLAB based on HMM and GMM. Thus, the advantages of using those techniques in speech recognition can be achieved.

Applying MFCCs with its 12 features per frame was not enough to train the HMM and hence, energy and deltas were calculated as an enhancement step. The energy features was improved using GMM to train HMM and produce a good recognition rate. Deltas features are excluded to reduce the processing time, as its impact was ineffective on the accuracy. Using GMM instead of K-means before HMM training improved the recognition rate from sixties to nineties.

In this work, twelve persons (males and females)
recorded 5 sentences with different length between 5 and 8 words in each sentence. Each person used his own mobile device to record with different circumstances. Accuracy of 94% is achieved by using HMM and GMM. This recognition rate is achieved either testing separate phrases or more phrase in one sound file. Comparisons are presented to measure the acceptability and intelligibility of our system.

The rest of the paper is organized as follows: Section II presents the related work while Sections III and IV present the HMM background and Arabic speech recognition system respectively. The implementation and results are presented in section V. Comparative analysis of the proposed system and state of the art are presented in Section VI. The paper ends with conclusions and suggestions for further work.

II. RELATED WORK

Many ASR had been proposed in this hot research area, some of them used HMMs [11] in recognition and the others used neural network [14], statistics [2], or hybrid techniques [15]. Our target is to use HMM in recognition step as it is the most common, widely used and achieved excellent results in the area of speech recognition.

Real time ASR system is proposed [2] using MFCCs and Euclidean Distance. The authors used pre-emphasis, framing, and windowing for preprocessing the audio signals that they acquired from four persons. Then, extract the features of the processed speech using MFCCs. Euclidean Distances are calculated for the features vector of the unknown letter and the stored letters in the database then the letter with the lowest Euclidean Distance is the recognized letter. This system recognized letters only not words or phrases but they achieved a good recognition rate of 89.6%.

Dynamic Time Warping (DTW) [8] is used as a feature matching techniques by comparing the tested voice with the stored template database to get the similarity measurements. Warping function is produced by DTW to diminish the total distance between the respective points of the signal. MFCCs are used for the feature extraction after using some preprocessing algorithms like pre-emphasis, framing, Hamming windowing, Fast Fourier Transform (FFT), and Mel filter bank processing.

Russian speech recognition system [11] is build using a HTK toolkit that is developed and tested by Cambridge University. The Russian system used the speech recognition principles and algorithms to establish Russian acoustic and language models. The system consists of training and recognition phases. Parameters adjustment and evaluation of the HMM model are done in the training phase. The best matching process is found in the recognition phase based on the recognition network that established from the existing HMM model, data dictionary, and grammar. The result of testing an internal speech was high recognition rate but unfortunately low recognition rate was achieved for testing an external speech. They recommended improving the system using other algorithms and techniques and also increase the dataset.

MFCC, Gaussian Mixture model (GMM) and Linear Predictive coding (LPC) are used in the automatic speech recognition technique for Bangla words [16] to extract the features of 100 words and each word repeated 10 times. Matching between feature matrix and reference matrix are done using the dynamic time warping, posterior probability function, and Euclidian distance. The decision is taken based on the distance measurements between feature matrix and all
reference matrixes. The reference matrix with the minimum distance with the features matrix (tested word) is chosen. A combination of the used techniques produced four different models. The results showed that the model using MFCC and GMM for feature extraction and posterior probability function for matching has the highest recognition rate of 84%.

Neural Network Language Model (NNLM) and Structured OUtput Layer (SOUL) are used [5] as an ASR system that evaluated using Mandarin Chinese and Arabic data. The SOUL approach used the clustering tree to build the vocabulary structure from the continuous speech by integrate the advantages of classifier-based language models and neural networks. NNLM built a real-valued feature vector for each word in the continuous speech forms and the similar words are distributed as neighbors in the continuous space. These similarities between words are considered in the smooth function of the n-gram distributions. The word forms in the continuous space and the associated probabilities are estimated using neural network. The performance of the SOUL NNLMs achieved significant improvement in the recognition rate compare to shortlist NNLMs interpolated with conventional 4-gram LMs.

An Arabic speech recognition system is built based on CMU Sphinx System [3]. Acoustic and language models with Arabic speech data are generated and trained in the system. Arabic characters are used to build an adopted dictionary. Some states of HMM, some Gaussians Densities, word insertion probability silence, insertion probability, and language weight adjustment are altered in the CMU Sphinx tool to make it appropriate for Arabic system and also some new scripts are added. MFCCs are used to extract features then the system is trained iteratively using Baum-Welch or forward-backward training algorithm. The system is tested using a new dataset and achieved a good recognition rate but it can be improved using more dataset. Another system that used CMU Sphinx system is implemented to test Arabic Digits and achieved a good recognition rate [5].

The work presented in paper [2] is the closest to our work. It is an ASR system for Arabic language based on MFCCs but they used Euclidean Distances calculation in recognition stage and also the system is for recognized some Arabic letters. Our work deals with continuous Arabic speech in its formal form. In addition, the Arabic ASR system is based on MFCCs and HMM with GMM.

III. HMM BACKGROUND

Hidden Markov Model (HMM) is widely used in many application especially speech recognition systems. This is because it has a rich mathematical structures and it works well in practical implementation for many applications. The Russian mathematician Andrey Andreyevich Markov is the owner of stochastic Markov processes theory that known later with Markov process and Markov chains [17].

HMM is a finite learnable stochastic automate that consists of a hidden variables (states) and observable outcomes. It is known as a doubly embedded stochastic process. 1) The underlying stochastic process is the finite set of states where the probability destination of each state is multidimensional associated with the other states and the transition probabilities are responsible of states transition organization. 2) The other stochastic process is the set of outcomes from any state that is not observable [17].
Five variables are needed to determine HMM: the number of states in the model, the state probabilities that define the number of outcomes for each state, the transition probabilities that describe connection structure of HMM, the emission probability for each state that define the probability of predetermine outcome is emitted form a specific state, and the initial probabilities that specify the probability of the model to be in a specific state at time zero. If the model has continuous outcomes then the number of outcomes of the state is infinite and continuous probability density function is used to calculate the emission probabilities instead of a set of discrete probabilities [17].

The common notation for HMM model that has \( N \) states and \( M \) outcomes is stated as in (1). Where \( A \) is the State transition probability distribution, \( B \) is the Observation symbol probability distribution in each state and \( \pi \) is the initial state distribution of the model [18].

\[
\lambda = (A, B, \pi)
\]  

(1)

The evaluation, decoding, and learning problems are the three basic problems that can be solved using HMM. The evaluation problem states the probability of generating specific outcomes given the HMM model \( \lambda \) that can be used in word recognition application. The decoding problem determines the state sequence of the HMM model \( \hat{\lambda} \) that can generate the specific outcomes to be used in continuous recognition application. The learning problem is the adjustment of \( \{A, B, \pi\} \) parameters to maximize the probability of a given outcomes and a given HMM model \( \hat{\lambda} \) that can be useful in training HMM model for recognition tasks [17]. The last one is the best choice for creating a good model for real applications [18].

The learning problem is the hardest issue used in this paper as we prepared continuous speech files to train HMM model to build an Arabic speech recognition system and evaluate it. The Baum-Welch algorithm and gradient techniques are used to choose the parameters of the HMM to get the specific sequence of outcomes (training). The Baum-Welch algorithm is the most commonly used and it combines between expectation-maximization algorithm and forward-backward algorithm [18].

### IV. ARABIC SPEECH RECOGNITION SYSTEM

The automatic continuous speech recognition system for formal Arabic language block diagram is shown in Fig. 2. It consists of four main steps dataset preparation (audio files collection), preprocessing, feature extraction, HMM training, and system evaluation using different audio files. MATLAB is used to implement the system using MFCCs and HMM.
Fig. 2 Automatic continuous speech recognition system for formal Arabic language

A. Database Set Preparation

The entire document should be in Times New Roman or Times font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes.

Arabic language has many dialects like Egyptian Jordanian, Kuwaiti, Emirati, Lebanese, and many others. All Arabic natives know and understand the formal Arabic but sometimes they cannot understand each other. Therefore, we choose to train our system with formal Arabic speech audios taken from the news channels like Aljazeera channel [19].

We randomly pick five sentences from Aljazeera channel news files. These sentences length are different between 5 and 8 words. The total number of words feed to the system is 30 words. Twelve persons including males and females record these sentences five times for each sentence. Each person used his own mobile phone recorder to record the required sentences that means that the audio record circumstances are different. Audios duration are varied between 2 and 8 seconds according the pronunciation speed of the person and the sentence length. The collected speeches are converted to *.wav forms to be ready for the next step preprocessing and features extractions. A summary of dataset is shown in TABLE I.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>DATASET SUMMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of sentences</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Sentence 1 : 6 words</td>
</tr>
<tr>
<td></td>
<td>Sentence 2 : 5 words</td>
</tr>
<tr>
<td></td>
<td>Sentence 3 : 5 words</td>
</tr>
<tr>
<td></td>
<td>Sentence 4 : 6 words</td>
</tr>
<tr>
<td></td>
<td>Sentence 5 : 8 words</td>
</tr>
<tr>
<td>Total no of words</td>
<td>30 words</td>
</tr>
<tr>
<td>speech duration</td>
<td>2-8 seconds</td>
</tr>
<tr>
<td>No of person</td>
<td>12</td>
</tr>
<tr>
<td>Person sex</td>
<td>10 females an 2 males</td>
</tr>
<tr>
<td>Recording device</td>
<td>Different smart phones</td>
</tr>
<tr>
<td>No of samples/sentence</td>
<td>60 files</td>
</tr>
<tr>
<td>Dataset size</td>
<td>300 files</td>
</tr>
<tr>
<td>Training set</td>
<td>250 files</td>
</tr>
<tr>
<td>Testing set</td>
<td>50 files</td>
</tr>
</tbody>
</table>

B. Preprocessing and Features Extraction

Preprocessing the audio files is important step to prepare the files for the feature extraction step. MFCC is used to extract good features that can be helpful in recognition step using HMM to achieve a good recognition rate.

MFCC mimics some of the human hearing that has limited perceptions for frequencies over 1Khz. MFCC is based on two types of filter: the linear filter for the low frequencies (<1000 Hz) and the logarithmic filter that perceive loudness and pitch of human acoustic and eliminates the frequencies and harmonics of the speaker characteristics [8]. The block diagram of the used MFCC technique is shown in Fig. 3. It can be classified into preprocessing and feature extraction steps then these two main steps can be summarized in the following steps:

1) Read audio files and convert it to digital form using Analogue to Digital Converter (ADC).
2) Apply pre-emphasis higher frequencies filter to the audio signals.
3) Divide the audio signal into small frames (20-40 ms).
4) Apply Hamming window.
5) Apply Fast Fourier Transform (FFT).
6) Apply the Mel filter bank using Mel scale.
7) Take the logarithm of all filter bank energies.
8) Apply DCT to extract 12 MFCC coefficients.
9) Calculate the energy feature then add it to the features vector of each frame.

1. Preprocessing
Our files are continuous time signals. We need to convert the continuous time signal $X(t)$ to the discrete time representation using a periodic sampling. The length of analogue signal and the sampling frequency are the factors that specify the sample size using (2):

$$X[n] = X(nT)$$  \hspace{1cm} (2)

Where $n$ is the number of samples, $T$ is the sampling period, and $X[n]$ is the sequence of samples.

The low frequency signals have more energies or amplitudes than the high frequency signals. Pre-emphasis higher frequencies filter is used to boost the energy of the higher frequencies. This step is optional as it can be done through next steps like channel normalization technique in Mel filter bank.

Framing the digital audio signals into 20-40 ms frames is an important step to get a reliable spectral estimate. This is based on the assumption that the audio signal does not change statistically much over the short time scales compared to many changes over the long time scales.

Hamming window is the commonly used window shape in speech recognition systems especially in MFCC technique because it diminishes the signal values at the window boundaries and avoids discontinuities. The main advantage of Hamming window is the next block consideration in the step of features extraction chain and the integration of all the closest frequency lines. If we consider that $N$ is the number of samples in each frame, $Y(n)$ is the output signal, $X(n)$ is the input signal, and $W(n)$ is the Hamming window, so (3) and (4) describe the Hamming window and its effect on the signal [20]:

$$Y(n) = X(n)*W(n)$$  \hspace{1cm} (3)

$$W(n)=\begin{cases} 
0.54 - 0.46 \cos \left( \frac{2\pi n}{N-1} \right) & 0 < n < N - 1 \\
0 & \text{otherwise}
\end{cases}$$  \hspace{1cm} (4)

The information of the speech signal in the frequency domain is much more than the information in the time domain. Identifying the frequencies in the frame is done using FFT by transforming the signal from the time domain to the frequency domain [20] using (5). Where $X(f)$, $H(f)$ and $Y(f)$ are the Fourier Transform of $X(t)$, $H(t)$ and $Y(t)$, respectively.

$$Y(f)=\text{FFT}[H(t)*X(t)] = H(f) \cdot X(f)$$  \hspace{1cm} (5)

2. Feature Extraction
Mel-scale frequency is used instead of Hertz to mimic the human perceptions simulation. Mel filter bank is used to get an idea of how much energy exists in various frequency regions. Two triangle filters are used: the very narrow filter indicates the energy amount exists near 0 Hertz and the wider filter indicates the energy amount at each spot in high frequencies. Mel scale answers the questions of how to space the filter banks and what is the width of them using (6):
\[ F(\text{Mel}) = [2595 \times \log_{10} [1+f] 700] \]  
(6)

We do not hear loudness on a linear scale. The logarithm allows us to use cepstral mean subtraction of Mel filter bank energies, which is a channel normalization technique. In addition, logarithm makes the feature estimation less sensitive to variations in input in other words speaker independent [20].

Acoustic vectors are the MFCCs that result from converting logarithm Mel spectrum into the time domain from the frequency domain, using DCT. Only 12 DCT coefficients are kept and the remaining of 26 is discarded to improve the speech recognition system.

Feature extraction is almost done after applying DCT but the features vector has only information about power spectral envelope of a single frame. We need to add more features to improve the system and increase the recognition performance as the features related to the energy and the change in cepstral features over time. Energy of the signal \( X \) from time sample to another in a frame is computed as stated in (7).

\[
\text{Energy} = \sum X^2(t) 
\]  
(7)

In this research, we used 12 cepstral features and energy feature for each frame which means that we have a feature vector with dimension 13.

V. IMPLEMENTATION and RESULTS

A. Implementation

The system implementation is done using MATLAB R2012a within an HP Pavilion dv6 Notebook PC with the following characteristics: Intel(R) core (TM) i3 CPU M350 @ 2.27GHZ 2.27 GHZ, 4GB RAM, and Windows 7 home basic service pack1.

The HMM toolkit for MATLAB is used in our implementation which supports the interface and learning of HMM and many algorithms like discrete outputs, Gaussian outputs, mixtures of Gaussians output, and others [21].

Using GMM with HMM is considered as a post-processing step for the audio files. GMM parameters: mean, variance and mixtures weight are estimated using Expectation Maximization (EM) algorithm for the spectral histogram of the FFT. GMM is fast, easy to learn and, has a reliable training [22].

The used dataset for this research is collected from different persons in formal Arabic. Five sentences repeated sixty times. Three hundred files are randomly divided as two hundred fifty for training and fifty for testing.

We can summarize the steps of the system as follow:
1) Reading the audio files categorized into 5.
2) Compute MFCC features with 13 coefficients for each word samples (60).
3) GMM is applied to compute mean, variance and mixtures weight.
4) Training: three iteration of EM is used to improve HMM parameters for each sentence and its dataset.
5) Testing: find the log likelihood for the parameters of the testing audio with respect to all the training data.

B. Results

At first, the system is built using HMM for training and testing five isolated words and the achieved accuracy was 100%. This is done using the 12-feature vector resulting from DCT step and K-means clustering to initialize HMM instead of random initialization. Then we tried to use the same system for short sentences with two or three words only but unfortunately, we got bad recognition rate around 60%. We tried to tune the parameters to get the best
result and the recognition rate improved when we used GMM to initialize HMM instead of K-means clustering.

Using 12 features vector that result after applying DCT was enough for isolated word system to get 100% recognition rate. However, for continuous speech those 12 features did not satisfy an acceptable recognition rate. Our target was to get a recognition rate more than 90%. After adding the energy feature to the features vector, we achieved better recognition rate after enough training with three iteration only of EM to improve HMM parameters.

As mentioned in TABLE I, the testing dataset containing 50 files, 10 samples for each sentence. The last 10 samples of each sentence are reserved for testing. Those 10 samples are from different persons as we randomly renamed all samples of each sentence at beginning. The recognition rate achieved for the testing file is 94% as shown in TABLE II.

TABLE II
TESTING RESULTS OF SEPARATED FILES

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Sample</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentence 1</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Sentence 2</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Sentence 3</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>90%</td>
</tr>
<tr>
<td>Sentence 4</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Sentence 5</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>80%</td>
</tr>
<tr>
<td>Total</td>
<td>50</td>
<td>46</td>
<td>4</td>
<td>94%</td>
</tr>
</tbody>
</table>

Another testing is done using 7 audio files, each one contain the five sentence but in different orders. As example, the second sentence is recorded then first sentence then third then fifth and the last one is the fourth sentence. Based on the silence, each file is divided into five files and tested. The result of such step is shown in TABLE III. We note that the recognition rate ranges between 80 and 100% with an average recognition rate of 91.4%.

TABLE III
TESTING RESULTS OF CONTINUOUS FILES

<table>
<thead>
<tr>
<th>File</th>
<th>Sentence order</th>
<th>Result order</th>
<th>Incorrect</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>File 1</td>
<td>[2 1 4 5]</td>
<td>[2 1 4]</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>File 2</td>
<td>[3 4 2 5]</td>
<td>[3 4 2]</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>File 3</td>
<td>[2 4 1 3]</td>
<td>[2 4 1]</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>File 4</td>
<td>[5 4 3 2]</td>
<td>[5 4 3]</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>File 5</td>
<td>[1 2 3 5]</td>
<td>[2 2 3]</td>
<td>1</td>
<td>80%</td>
</tr>
<tr>
<td>File 6</td>
<td>[2 4 1 3]</td>
<td>[2 4 1]</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>File 7</td>
<td>[1 4 5 3]</td>
<td>[1 1 5]</td>
<td>1</td>
<td>80%</td>
</tr>
</tbody>
</table>

VI. COMPARATIVE ANALYSIS
Comparing our system results with the state of the art works is considered as a second step of system evaluation. Most of the published papers that present an ASR for Arabic language are summarized in TABLE IV.

As we notice from TABLE IV, most of the work concerned with the isolated word recognition system [14], [16], [24] as it is much easier than continuous
speech and can build large database from isolated words easily.

The results in [1] used 3000 letters database to recognize some isolated words and they achieved a good recognition rate. The work in [2], [7] used the letters dataset to recognize letters while [14], [16, [24] used words dataset to recognize the words. All the above mentioned works implemented the ASR for isolated words with different recognition techniques. Paper [23] is the only one that implemented ASR for continuous speech but they used two types of dataset; the first dataset (2110) for isolated words and the second dataset (415) for continuous speech.

Our previous trial to build formal Arabic ASR system for isolated words achieved 100% recognition rate. However, for continuous speech we achieved better recognition rate of 94% compared to the work published in [23]. Based on this comparison, our proposed system has a good acceptability and intelligibility.

<table>
<thead>
<tr>
<th>Table IV</th>
<th>COMPARISON BETWEEN ARABIC ASR WORKS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Published Work</td>
</tr>
<tr>
<td>Paper [1]</td>
<td>Words</td>
</tr>
<tr>
<td>Paper [2]</td>
<td>Letters</td>
</tr>
<tr>
<td>Paper [7]</td>
<td>Letters</td>
</tr>
<tr>
<td>Paper [14]</td>
<td>Words</td>
</tr>
<tr>
<td>Paper [16]</td>
<td>Words</td>
</tr>
<tr>
<td>Paper [23]</td>
<td>Continuous Speech/Words</td>
</tr>
<tr>
<td>Paper [24]</td>
<td>Words</td>
</tr>
<tr>
<td>Our Work</td>
<td>Continuous Speech</td>
</tr>
</tbody>
</table>

VII. CONCLUSION

A continuous formal Arabic Speech Recognition system using MFCCs and HMM with GMM is presented and evaluated. The Arabic speech recognition system is built using MATLAB based on HMM toolbox. A recognition rate of 94% is achieved after adding energy feature for each frame to the MFCCs features vector and using GMM to initialize the HMM instead of K-means. A comparison between our work and the state of the art works showed that our system is acceptable and intelligent enough. Many trials are done before achieving this level of acceptability and intelligence. Arabic language speech recognition system is not yet mature and it needs more developments because the Arabic spoken language is one of the most broadly used languages of the world.

Increasing dataset can led to better results but collecting a specific data like the data used in this research, is not an easy work and need a lot of time and efforts. Therefore, we will concentrate our future work on four points. The first one is to increase the number of sentences with different length. The second is to acquiring more dataset. The third is to improve the system to reduce the time of training to get a faster system. The fourth and last one is to train the system with paragraphs; each paragraph consists
of some sentences.

VIII. ACKNOWLEDGMENT
This work was supported and funded by Kuwait University, Research Project No. (QE 06/14).

IX. REFERENCES


