American Journal of Applied Sciences 4 (1): 23-32, 2007
ISSN 1546-9239

CASRA+: A Colloquial Arabic Speech Recognition Application

Ramzi A. Haraty and Omar El Ariss
Lebanese American University, Beirut, Lebanon

Abstract: The research proposed here was for an Arabic speech recognition application, concentrating on the Lebanese dialect. The system starts by sampling the speech, which was the process of transforming the sound from analog to digital and then extracts the features by using the Mel-Frequency Cepstral Coefficients (MFCC). The extracted features are then compared with the system's stored model; in this case the stored model chosen was a phoneme-based model. This reference model differs from the direct word template matching, where speech features that are extracted from the input are directly compared to the word templates. Each word template in the direct matching model was stored as a vector of feature parameters. Thus, when the vocabulary size of the ASR system becomes large, the memory size for the word template will become humongous. In contrast, the model used here was phoneme-like template matching. Word templates are stored as phoneme-like template parameters. Thus, the memory size for the word templates will not grow as fast as that of the direct matching model.

Key words: Arabic language, speech recognition and template matching

INTRODUCTION

The speech wave itself contains linguistic information that includes the meaning the speaker wishes to impart, the speaker's vocal characteristics and the speaker's emotion. Speech recognition is the process of automatically extracting and determining linguistic information conveyed by a speech wave using computers or electronic circuits. Only the linguistic information is needed from the speech wave, while the rest of the information is used in other fields of signal processing. A speech recognition system performs three primary tasks as shown in Fig. 1.

* Preprocessing: converts the spoken input into a form the recognizer can process.
* Recognition: identifies what has been said by comparing the input with the built-in reference models.
* Communication: sends the recognized input to the software systems that needs it.

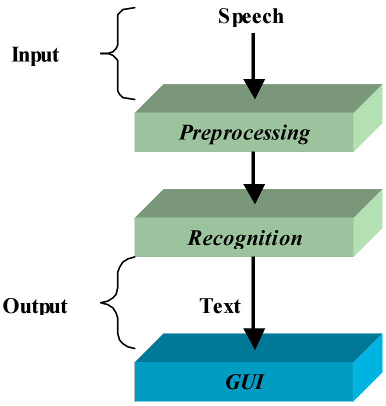

Fig. 1: The structure of a speech recognition system

Speech recognition through computer software encounters diverse types of difficulties due to the enormous information that is carried with the speech signal. Therefore, the need to apply constraints to simplify the difficulties is needed in order to make the recognition process possible. Some of the constraints could be the recognition of isolated words, limitation in the vocabulary size, or a limitation in the number of speakers. For example, as a constraint this system accepts isolated words as input in order to make the process of endpoint detection (word spotting) much easier. Some of the difficulties encountered by a speech recognition system that are related to the Arabic language are:

Word knowledge: Speech is not just acoustic sound patterns, additional knowledge, as word meanings, is needed in order to recognize exactly the intended speech. Therefore, words with widely different meanings may share the same sequence of sound patterns. For example:

The word'كَلَّ' that means exhausted and the word 'كَلاَّ' that means no or never.

The word 'جَرَّ' that means to drag, the word 'جَرَّى' that means to make something to stream and the word 'جَرَّة' that means a jar.

Variability caused by dialectical differences: Variability in dialect between Arab countries and even dialectical difference in the same country causes the word to be pronounced in a different way. This variability in word pronunciation might cause an error in recognition. An example of dialectical difference

Corresponding Author: Ramzi A. Haraty, Lebanese American University, Beirut, Lebanon

between Arab countries: speakers in Egypt pronounce the phoneme ‘ج’ in the word ‘جمال’ as the letter g in ‘get’, while speakers in Lebanon pronounce the phoneme similar to the letter j in ‘jar’. An example of dialectical difference in the same country: some of the people that live in Beirut spell the word ‘أنا’ as ‘أني’.

Coarticulation effects: The acoustic realization of a phoneme may heavily depend on the acoustic context in which it occurs. This effect is usually called coarticulation. Thus, the acoustic feature of a phoneme is affected by the neighboring phonemes, the position of a phoneme in a word and the position of this word in a sentence. Such acoustic features are very different from those of isolated phonemes, since the articulatory organs do not move as much in continuous speech as in isolated utterances. We can see the effect of coarticulation in the following phrase ‘و في الأيام’. Here the phoneme ‘ي’ in the word ‘في’ is affected by the neighboring phoneme ‘فِ’ and by the phoneme ‘لْ’ in the word ‘الأيام’. Therefore the acoustic realization is different from the stand alone phoneme ‘ي’.

Diacritization: Diacritics that are described in the section 3 play an important part in written Arabic material. The absence of diacritics in most Arabic texts causes many ambiguities in the pronunciation of words. Therefore, a speaker using an Automatic Speech Recognition (ASR) system while reading form a non-diacritized source might cause him to mispronounce some words thus causing errors in recognition. Some of the diacritic variation for the word ‘رحم' are: 'رَحِمَ', 'رُحِمَ', 'رَحَّمَ','رَحِمْ'.

Morphology: The Arabic language is morphologically rich, thus causing a high vocabulary growth rate. This high growth rate is problematic for language models by causing a large number of out-of-vocabulary words. Papers[1,2] address the effect of morphology on Arabic language speech recognition systems.

Although there was and still continues, extensive research and advancements in speech recognition on English language, there has been little research done on the Arabic language. In addition to that, most of the research done is either for the standard (formal) Arabic language or the Egyptian colloquial language. The reason for this shortage in research is due to the diversity of colloquial Arabic, the differences that exist between one colloquial type and another and the lack of written material for the colloquial Arabic. This paper describes the implementation of an isolated word recognition system using pseudo-phoneme (phoneme-like) templates based on the Lebanese colloquial Arabic dialect.

* The increase of research in the field of automatic speech recognition is due to the fact that implementing computer software that supports speech brings with it many advantages[3,4]:
* Speech input is easy to perform because it does not require a specialized skill as typing or pushbutton operations.
* Speech can be used to input information three to four times faster than typewriters and eight to ten times faster than handwriting.
* Information can be input even when the user is moving or doing other activities involving the hands, legs, eyes, or ears.
* Speech as input is more suitable for individuals challenged with a variety of physical disabilities, such as loss of sight or limitations in physical motion and motor skills.

These advantages of speech recognition are what to be sought for in Arabic language software. Research done on Arabic ASR is not many and previous work can be divided into two groups. The first group focused on the recognition of formal (standard) Arabic. The first formal Arabic speech recognition system was the BBN Tides On Tap system[5,6]. The second work, an isolated speech recognition based on a hybrid Hidden Markov Model (HMM) was implemented by[7]. The third work, an isolated speech recognition system based on Neural Network was done at the American University of Beirut[8]. Finally, another Neural Network system that also accepts isolated speech was developed by[9]. The second group focused on the recognition of Egyptian Colloquial Arabic (ECA). The first recognizer for colloquial Arabic was developed by BBN[10,11]. The system focused on recognizing the Egyptian dialect in addition to the English and Spanish language. Recent works that based there testing on the same ECA CallHome corpus were researched by[1,2,12]. Papers[1,2] focused on the improvement of the morphological aspect of the language model, while[12] focused on the cross-dialectical data using standard Arabic to improve the recognition rate of ECA.

The Arabic Language: Linguistically speaking, Arabic language does not have a normalized form that is used in all circumstances of speech and writing. Arabic used in daily informal communication is not the same form of Arabic that is used in books, magazines, newspapers and on TV to broadcast the news. While written Arabic in text materials is standardized and is the same in the entire Arab world, there is no standardization for Arabic that is spoken informally. This lack of standardization and lack of rules caused the spoken Arabic to be considerably varietal from one region to another. The forms of Arabic are as follows:

Classical or formal Arabic: is the old form of the language. It can be seen in the Jahelia poetry.

Modern Standard Arabic (MSA): is a version of classical Arabic with modernized vocabulary. It is considered to be the formal language that is common in all Arabic speaking countries. Modern Standard Arabic is the form of Arabic used in all written texts.

Colloquial or dialectical Arabic: There are many different dialects that differ considerably from each other and from the Modern Standard Arabic. According to[2], colloquial Arabic can be divided into two groups: Western Arabic and Eastern Arabic. Western Arabic can be subdivided into Moroccan, Tunisian, Algerian and Libyan dialects. While Eastern Arabic can be subdivided into Egyptian, Gulf and Levantine (Iraqi, Jordanian, Lebanese and Syrian) dialects. This categorization is considered to be loose due to the fact that dialectical differences are not the sole product of the differences in country. Other factors such as rural or urban regions and tribal play important roles in the way the dialect is formed. Dialectical forms of Arabic can be many even in one country. For example in Lebanon, dialects are different in the south, north, Beirut and the mountains, further dialectical subdivisions can also be made. Another example, in Oman the dialect spoken is similar to the dialect spoken in Sudan and not to the other Gulf countries. The regional dialects of Arabic are spoken languages; very little written dialectical material exists.

Although some consider the alphabet to consist of twenty-eight letters (excluding the hamza)[2,13], the Arabic alphabet consists of twenty-nine letters, shown in Table 1. Additional symbols or letters can be introduced for certain phones that are not present in the Arabic alphabet (like the English phonemes [p] and [v]).

Arabic doesn't have letters for vowels; all the alphabets are consonants. Diacritics play an important role in forming short vowels. The fatha, kasra, damma and tanween all form different short vowels for the same letter. Long vowels can also be produced by adding an 'ا' after a short vowel. Also the madda diacritic form a long vowel for the letter 'ا'. The sokoon means that the letter is a consonant, while the shadda doubles the letter (the first is a consonant while the other letter is a vowel). Although diacritics play an important role in the way a written Arabic is pronounced by adding vowels to the language, most of the written texts are not diactritized. The lack of diacritization of Arabic texts can be compared to an English text in which the vowels are removed from it. This lack of diacritics in a word might cause considerable ambiguities, leading the speech recognition system process to give wrong results. The word 'رحم' as an example has at least five possible diacritizations. Therefore in order for a template-based speech recognition system to recognize those diacritization, the system must have at least one model for every diacritization form. Table 2 lists all the Arabic diacritics:

The Modern Standard Arabic has at least one hundred twelve phonemes. Every letter except the letter 'ا', which is not included because it just changes the vowel duration from short to long, are affected by the four diacritics: fatha, damma, kasra, sokoon. Therefore, every letter has four phonemes.

Lebanese Colloquial Arabic: Lebanese colloquial Arabic is the spoken Arabic used by the Lebanese people in oral communication. We will refer in this paper to the common characteristics of the different dialects in Lebanon as the Lebanese colloquial Arabic. This dialect has some differences compared to the other dialects of the Arab world and to the standard Arabic. These differences exist in all the levels of the language through pronunciation, phonology, meaning, morphology and syntax. So some phonemes are replaced by other phonemes, some words are pronounced differently, some words have the same pronunciation but a different meaning and some words are unique to this dialect. Although in some of the regions in Lebanon the differences in dialect are strong, but a common structure between those dialects is viable. Some of the characteristics of the Lebanese dialect when compared to standard Arabic are stated below[14]:

The letter 'ء' (hamza): In many cases were the hamza occurs in the beginning or in the middle of the word, it is dropped. For example 'إكسر' becomes 'كسور' and 'رأس' becomes 'راس'.

In some cases the hamza is replaced by the letter 'ي'. For example 'بئر' becomes 'بير' and 'مصائب' becomes 'مصايب'.

In some words the hamza is replaced by the double letter 'ي'. For example 'مِئة' becomes 'مِيّة'.

The letter 'ثـ' (thaa): In most of the cases when it occurs in the beginning or in the middle of a word, the thaa is changed into the letter 'تـ'. For example 'ثور' becomes 'تور' and 'إثنين' becomes 'تنين'.

When the thaa occurs at the end of the word, most of the time it is changed into the letter 'س'. For example 'حديث' becomes 'حديس' and 'خبيث' becomes 'خبيس'.

The letter 'ذ' (thal): This letter is always replaced by either the letter 'د' or the letter 'ز'. For example 'ذهَبْ' becomes 'دهَبْ', 'إذا' becomes 'إزا' and 'هذا' becomes 'هيدا'.

The letter 'ظـ' (zaa): Many times this letters is replaced by the letter 'ض'. For example 'ظفُرْ' becomes 'ضفُرْ'.

Table 1: The arabic alphabet

Letter	Name	Consonant (sokoon)	Short vowel (fatha)	Short vowel (damma)	Short vowel (kasra)	Long vowel (ا)	Long vowel (و)	Long vowel (ي)
ب	Baa	بْـ	بَـ	بُـ	بِـ	با	بُو	بِيـ
ت	Taa	تْـ	تَـ	تُـ	تِـ	تا	تُو	تِيـ
ث	Thaa	ثْـ	ثَـ	ثُـ	ثِـ	ثا	ثُو	ثِيـ
ج	Gym	جْـ	جَـ	جُـ	جِـ	جا	جُو	جِيـ
ح	Haa	حْـ	حَـ	حُـ	حِـ	حا	حُو	حِيـ
خ	Khaa	خْـ	خَـ	خُـ	خِـ	خا	خُو	خِيـ
د	Daal	دْ	دَ	دُ	دِ	دَا	دُو	دِيـ
ذ	Thal	ذْ	ذَ	ذُ	ذِ	ذا	ذُو	ذِيـ
ز	Zayn	زْ	زَ	زُ	زِ	زا	زُو	زِيـ
ر	Raa	رْ	رَ	رُ	رِ	را	رُو	رِيـ
س	Seen	سْـ	سَـ	سُـ	سِـ	سا	سُو	سِيـ
ش	Sheen	شْـ	شَـ	شُـ	شِـ	شا	شُو	شِيـ
ص	Saad	صْـ	صَـ	صُـ	صِـ	صا	صُو	صِيـ
ض	Daad	ضْـ	ضَـ	ضُـ	ضِـ	ضا	ضُو	ضِيـ
ط	Taa	طْ	طَ	طُ	طِ	طا	طُو	طِيـ
ظ	Zaa	ظْ	ظَ	ظُ	ظِ	ظا	ظُو	ظِيـ
ع	Ayn	عْـ	عَـ	عُـ	عِـ	عا	عُو	عِيـ
غ	Ghayn	غْـ	غَـ	غُـ	غِـ	غا	غُو	غِيـ
ك	Kaaf	كْـ	كَـ	كُـ	كِـ	كا	كُو	كِيـ
ق	Qaaf	قْـ	قَـ	قُـ	قِـ	قا	قُو	قِيـ
ف	Faa	فْـ	فَـ	فُـ	فِـ	فا	فُو	فِيـ
ل	Laam	لْـ	لَـ	لُـ	لِـ	لا	لُو	لِيـ
ن	Noon	نْـ	نَـ	نُـ	نِـ	نا	نُو	نِيـ
م	Meem	مْـ	مَـ	مُـ	مِـ	ما	مُو	مِيـ
ه	Haa	هْـ	هَـ	هُـ	هِـ	ها	هُو	هِيـ
و	Waw	وْ	وَ	وُ	وِ	وا	وُ	ويـ
ي	Yaa	يْـ	يَـ	يُـ	يـ	يا	يُو	يـ
ء	Hamza	ئْـ	أ	أُ	إ	آ	أُوُ	إيـ
ا	Alif	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Table 2: Arabic Diacritics

Symbol	Name	Meaning	Example
ْ	Sokoon	Consonant letter	حبْس
َ	Fatha	Short vowel	كَذَبَ
ُ	Damma	Short vowel	كُل
ِ	Kasra	Short vowel	عِند
ّ	Shadda	Letter doubling	شدّة
ً	Tanween el-fatha	Adds [an] to the letter	ايضاً
~	Madda	Turns the hamza into a long vowel	آدم

The letter 'ق' (qaaf): Most of the times this letter is changed into the letter 'ء' (hamza). For example 'طريق' becomes 'طريىء' and 'قال' becomes 'آل'.

Feature extraction: Figure 2 shows the structure of a speech signal analysis component in an Automatic

Speech Recognition system. The speech analysis, as shown below, can be summarized into three main stages, the first is done through hardware while the remaining two are implemented through software. The first stage can be shown as the movement of speech through the microphone, followed by the passage of the microphone output through the Analog to Digital (A/D) converter. The microphone transforms the pressure wave into an electrical analog signal, while the A/D converter digitizes or transforms the analog signal into a digital signal. The second stage is the extraction of the features from a digitized speech signal. The third stage recognizes the word uttered from the features extracted from the speech signal.

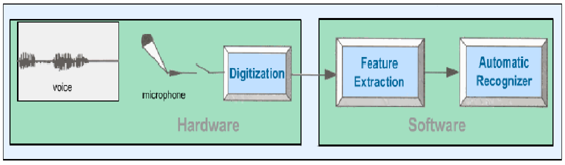

Fig. 2: Structure of a speech signal analysis

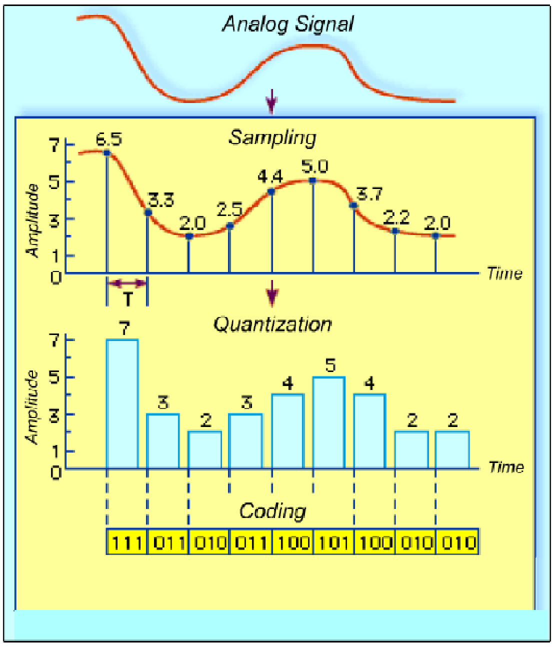

Fig. 3: Digitization of a speech signal domain (adapted from Encyclopedia Britannica , Copyright 2001)

Prior to feature extraction, the speech signal should be changed into digital form Therefore, capturing the speech wave is the first step to be done by a speech recognition system. The system starts by transforming the speech signal into a processable form, using a microphone, by converting it into an electrical signal. This electrical signal, which is an analog signal, is then changed into digital form using digitization. The reason for digitizing the speech signal is that digital techniques achieve a guaranteed accuracy and facilitate highly sophisticated signal processing which cannot be realized by analog techniques. The digitization process, is the process of converting the electrical speech signal into numerical values, could be done using special digital signal hardware, but in this research digitization is done using the audio sound card.

Without the use of digitization, the quantity of speech data would be so great that the processing and storage requirements would be prohibitive. In order for a speech recognition system to function at an acceptable speed, the amount of data must be reduced. Speech in addition to sounds contains also noise patterns and silences. Therefore, some data in the speech signal are redundant, some are irrelevant to the recognition process and some need to be removed from the signal because they interfere with accuracy of recognition. The challenge is to eliminate these useless components from the signal without losing or distorting critical information contained in the data, the process of digitization is shown in Fig. 3. This is done by choosing appropriate parameters for the digitization process[15]. Setting the parameters of the digitization process has a major effect on the relative error rate of the recognition process. A sampling rate of 16 kHz with a sampling precision of a 16-bit are chosen. That means, for every second the sound card returns 16,000 samples or numbers, each number is a double byte integer. The size of the memory buffer used for digitization is 4,096 double bytes or 8KB.

The extraction of reliable features is one of the most important issues in speech recognition. The Mel-Frequency Cepstrum Coefficient (MFCC) is chosen to be the feature extraction method due to the better performance and the ability of the frequency domain to model adequately the sound. The central theme is to decompose the speech signal into frames and then pass these frames into a linear time varying filter. The recognition system extracts acoustic patterns contained in each frame and captures the changes that occur as the signal shifts from one frame to the next. Figure 4 shows the components of an MFCC process with the number of input values for every component. The rest of this section will briefly define each component and how it is implemented.

Preemphasis: Formants, which are the peaks that result from the resonance of the vocal tract, usually define the structure of a phoneme. The high frequency formants carry with them relevant information, but they have smaller amplitude with respect to low frequency formants. Therefore, an amplitude that is the same for all formants should be attained. This can be done through the use of a Preemphasis filter, which flattens the spectral tilt. Preemphasis can be accomplished after the digitization of a speech signal through the application of the first-order Finite Impulse Response (FIR) filter[3,4]

$H(z) = 1 – \alpha z-1$

where α is the Preemphasis parameter set to a value close to 1, in this case 0.95. Applying the FIR filter to the speech signal, the preempahsized signal is related to the input signal by the relation:

$x'(n) = x(n) - \alpha x(n-1)$

here x' stands for the speech sample after Preemphasis, x the speech sample before Preemphasis and α the Preemphasis parameter.

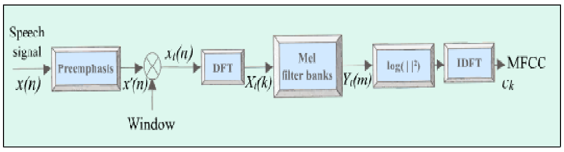

Fig. 4: Components of MFCC

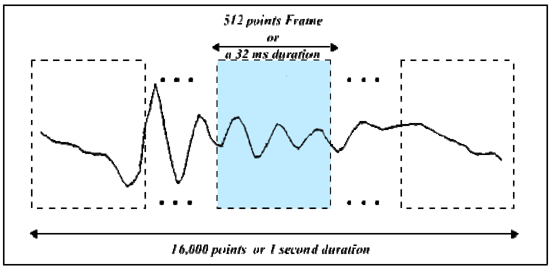
512 points Frame or a 32 ms duration
16,000 points or 1 second duration

Fig. 5: Frame segmentation of a speech signal

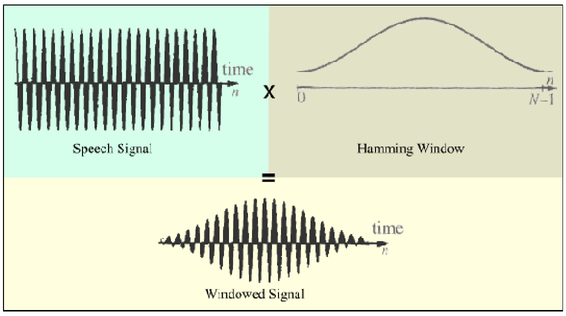

Fig. 6: Characteristics of a hamming window

Frame segmentation: Fourier transform, which will be discussed below, is reliable only when the signal is in a stationary position. For voice, this holds only within a short time interval usually less than 100 milliseconds. Therefore, the speech signal is decomposed into a series of short segments, called analysis frames, then each frame will be analyzed and useful features will be extracted from it. A 512 points frame, or approximately 30 millisecond duration, is chosen in this research, this frame segmentation can be seen in Fig. 5[3,4].

Windowing: To minimize the discontinuity and therefore preventing spectral leakage of a signal at the beginning and end of each frame, every frame is multiplied by a window function. Window functions are signals that are concentrated in time, often limited in duration, that consist of a central lobe which contains most of the energy of the window and side lobes which decay rapidly. There are many different window functions, like rectangular, hanning, hamming, triangular, Kaiser and many others, that can be applied to a speech signal. Here, the hamming window will be used. The characteristics and the application of this window to the speech signal can be seen in Fig. 6[3,4,16,17].

The hamming window is defined as:

$W_H(n) = 0.54 - 0.46 \cos(2n\pi/N-1)$ and the application of this window function to the speech signal is $x_t(n) = W_H(n).x'(n)$

where $x_t(n)$ stands for the speech sample at time n after applying the window function, W_H is the hamming window function and x' is the sampled speech after Preemphasis.

Fast fourier transform: Discrete Fourier Transform (DFT) is considered to be the basis of spectral analysis and spectral analysis reveals speech features that are due to the shape of the vocal tract. The Discrete Fourier Transform of a finite duration sequence {x(n)} where $0 \leq n \leq N - 1$ is defined as:

$X(k) = \sum x(n)e^{-j(2\pi/N)nk} = \sum x(n)W^{nk}$ where $(0 \leq n, k \leq N - 1)$

It can be easily seen that W^{nk} is periodic of period N and this periodicity is the key to the Fast Fourier Transform. The Fast Fourier Transform (FFT) is an algorithm that consists of variety of trick for reducing the computation time required to calculate a DFT. Although FFT algorithms are well known and widely used, they are rather intricate and often difficult to grasp due to the great variety of different FFT algorithms such as radix-4, split-radix, radix-8, radix-16 and decimation-in-time (DIF) algorithms[16,17].

This research implements the radix-2 algorithm. The idea behind this algorithm is to break the original N point sequence into two shorter sequences. This process continues by iterating, as long as N is an integer power of 2, until two point DFT's are left to be evaluated. The algorithm described here has been called the decimation-in-time (DIT) algorithm, since at each stage of the process, the input sequence is divided into smaller sequences; that is the input sequence is decimated at each stage[4,17].

Mel filter bank processing: This procedure has the role of smoothing the spectrum, closely modeling the sensitivity of the human ear. The Mel frequency scale is composed of a set of band-pass filters, generally 24 filters are used. The part of the spectrum which is below 1 kHz is usually processed by more filter banks since it contains more relevant information. Mel filters are linear below 1 kHz and logarithmic above, with equal numbers of samples taken below and above[4,16].

Log energy and IDFT: After smoothing the spectrum, the logarithm of the square magnitude of the coefficients is computed. The final step in MFCC consists of performing the Inverse Discrete Fourier Transform (IDFT) on the logarithm coefficients. The IDFT can be calculated using the FFT procedure.

Template matching: After feature extraction, the recognition process compares the extracted features with its reference model. In this research, templates are chosen as the reference modle. Template matching is a form of pattern recognition, where each word or phrase in an application is stored as a separate template. The input is then compared with the stored templates and the template that most closely match the incoming speech pattern is identified as the recognized word or phrase. The selected template is called the best match for the input. The representation is simple, straightforward and easy to generate, but it carries with it two drawbacks. The first, for every dialectical variance and diacritization of an Arabic word a distinct template should be included. For instance, a template for ‘أنا’ and another template for ‘أني’. The second, it is not good with recognizing words that require linguistic information. As an example, words that have similar sound (confusable words e.g., 'جَرَّة' and 'جَرَّ').

The template model used here is similar to the model used in the SPLIT system described in[3,18]. The system contains two template models. The first stands for phoneme-like templates, while the second is for word templates. Figure 7 depicts the block diagram of the system. First, phoneme-like templates are generated from the speech samples. The size of the samples used is about 48,500 vectors of speech features extracted as an output from the MFCC method. Vector quantization, which is described in the next section, is used to generate the 128 phoneme-like templates. Then word templates are generated. Here, each training word is divided into approximately a 30 milliseconds duration frame (same duration as the pseudo-phoneme’s and the MFCC frames) and compared to all the phoneme-like templates through a distortion measure. The phoneme-like template with the shortest distortion value represents the frame. In this way, each word template will be represented as a string, a vector, of phoneme like templates. In the recognition process, the word utterance is analyzed by the MFCC process outputting a vector of feature coefficients. These feature parameters are compared to every word template. Before the comparison happens, every phoneme-like template in the word template is replaced by its respective vector of features. The word template with the best comparison result to the input will be the word recognized. Dynamic time warping, described in section 7, is used to improve time-normalization during the comparison. The recognition process adapted here differs form the process implemented by the SPLIT system due to better recognition results. There, every frame of the input utterance is compared to all phoneme-like templates. Then the utterance is represented by phoneme like templates and compared directly with word templates.

Vector quantization: Quantization is the process of assigning discrete values to continuous amplitude signals. While quantization of a single parameter is called scalar quantization, joint quantization of multiple parameters is called Vector Quantization (VQ). The main goal of vector quantization is data compression, were a large size training data of speech vectors are

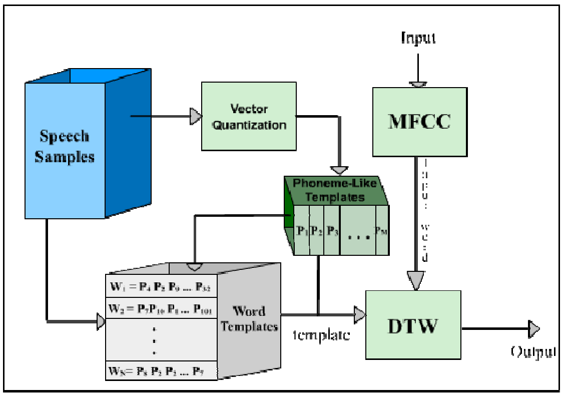

Fig. 7: Block diagram of the word recognition system

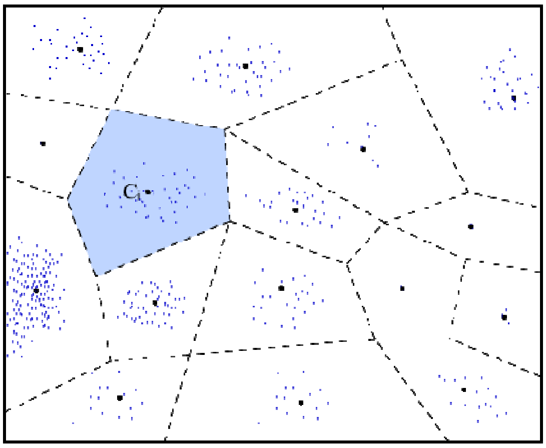

Fig. 8: An example of vector quantization

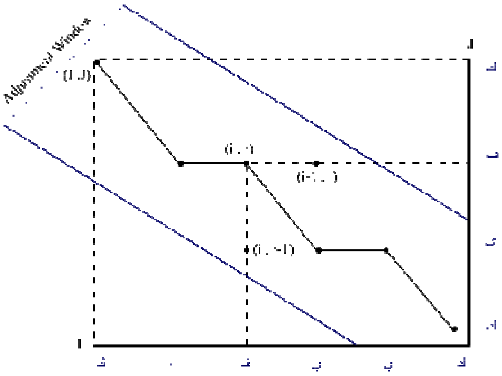

Fig. 9: Application of DTW

replaced by a smaller size of reproduction vectors (codevectors). Papers[19,20] give historical overview and a detailed description of the different ways to design a VQ model.

A VQ model is described by a codebook C, a partition space S and a quantization function q. The codebook $C = \{c_1, c_2, \ldots, c_N\}$ is composed of the dictionary of vectors, c_is, called codewords. N is the size of the codebook and is also referred to as the number of levels. The partition space $S = \{S_1, S_2, \ldots, S_N\}$ is the set of all encoding regions. The function of the quantizer q is to map (quantize) the input vector x, which is assumed to be k-dimensional into another k-dimensional (k here is defined as 13):

$$q(x) = c_i \qquad \text{if } x \, \varepsilon \, S_i$$

Figure 8 shows an example of a 2-dimensional vector quantization. Here, the two-dimensional space is divided into 16 regions or cells. Therefore, N = 16 and this implies that there are 16 codewords. In every region, the black point represents the centroid or the codeword while the small dots are the training vectors that belong to this region. The shaded cell, is the cell that the input x is mapped or quantized to. That means that the distance between x and c_i has the minimum, or the smallest, distortion measure between all other codewords where $1 \leq i \leq N$.

The first step in designing a vector quantization model is by building the codebook. Different methods can be used, like the K-means[21], to design the codebook. In this paper the LBG algorithm[19], or the splitting technique, is used. The algorithm works on a large training speech vectors of size M (in this case M = 48,500) to design the codebook that consists of 128 codewords. The algorithm starts by calculating the codeword for the whole M training vectors and then splits the initial codeword into two codewords. The algorithm iteratively resumes by calculating codewords then splitting them until the number of codewords reaches 128. The centroid or the codeword is calculated by the Euclidean center of gravity. The mapping of a training vector x to a certain codebook is done by using a distortion measure. There are many different types of distortion measures that can be used by the LBG algorithm. Paper[22] discusses the different types of distortion measures, while[23] compares the effect of different distortion measures on the design of a VQ model. Here, the squared-error distortion measure is used:

$d(x,y) = \sum |x - y|^2$ where both x and y are k-dimensional vectors

Dynamic time warping: Using word as a unit of recognition adds more complication to the recognition process. The speech signal is a time dependent process; therefore several utterances of the same word are likely to have different durations. Even the same word with the same duration might differ in the middle. The difference in duration is due to the different rates used while uttering the different parts of a word. For example, the word 'كيفك' has a different duration when the 'ي' is emphasized in the utterance 'كيييفك' and is also different in the utterance 'كيفااك' when 'فَ' is emphasized.

This problem can be solved through the use of Dynamic Time Warping (DTW). DTW nonlinearly expands or contracts the time axis to match the same phoneme position between the input speech, or the word uttered and the reference template. The DTW process can be efficiently accomplished by using the Dynamic Programming (DP) technique. The distance definition used here is a symmetric form of matching, were both the time axes of the input word and the word template are transformed into a temporarily defined common axis (i+j). The asymmetric type of matching and the comparison in performance to the symmetric type are covered in[24,25]. The DTW process can be represented mathematically as follows[3]:

$$g(i,j) = \min \left(\begin{array}{l} g(i, j-1) + d(i,j) \\ g(i-1, j-1) + 2d(i,j) \\ g(i-1, j) + d(i, j) \end{array} \right)$$

Here, d(i,j) is the local distance between i and j and g(i,j) stands for the global distance. An appropriate distance measure is the Euclidean distance. The total distance returned by DTW between the input word and the word template is:

$$D(\text{input word, template}) = g(I,J) * 1/(I + J)$$

An illustration of how DTW works is shown in Fig. 9. The x-axis represents the input word, while the y-axis represents the template reference. The uttered word ‘كييفاك’ is matched here with the template word ‘كيفك’. When there is no timing difference the warping function coincides with the diagonal line i = j, but deviates from it when the time difference becomes greater.

The first experiment, described in the next section, done to evaluate the system did not include slope constraints and the equation of DTW was as stated above. But in the rest of the experiments, slope constraints were applied. The constraint prevents the warping function from too steep or too gentle deviations. The intensity of the slope constraint can be evaluated by[24]:

$$P = n/m$$

Where n stands for the number of consecutive times a move is done towards the diagonal direction and m stands for the number of consecutive times a move is done towards the i or j axis. Here, P is set to 1 causing the DTW equation to become[24]:

$$g(i,j) = \min \left(\begin{array}{l} g(i-1, j-2) + 2d(i, j-1) + d(i,j) \\ g(i-1, j-1) + 2d(i,j) \\ g(i-2, j-1) + 2d(i-1,j) + d(i, j) \end{array} \right)$$

RESULTS AND DISCUSSION

Three tests were performed on the system in order to evaluate it.

Table 3: Test1 where the DTW has no slope constraint

Word	Articulation	# of Utterances	Correct	Wrong	%accuracy	%word error rate
أنا	أنا	20	20	0	100	0
سِتَة	سِتِّ	20	20	0	100	0
إثنَين	تَنِينْ	23	22	1	95.6	4.4
سَبْعِينْ	سَبْعِينْ	27	26	1	96.2	3.8
شُبَاطْ	شْبَاطْ	20	5	15	25	75
تِلْفَازْ	تَلْفِزْيُونْ	20	19	1	95	5
طَرَابْلُسْ	طْرابْلُسْ	28	27	1	96.4	3.6
Total		158	139	19	87.9	12.1

Table 4: Test results for 94 word utterances

Word	Articulation	# of Utterances	Correct	Wrong	%accuracy	%word error rate
أنا	أنا	15	15	0	100	0
سِتَة	سِتِّ	13	13	0	100	0
إثنَين	تَنِينْ	13	12	1	92.3	7.7
سَبْعِينْ	سَبْعِينْ	15	15	0	100	0
شُبَاطْ	شْبَاطْ	12	11	1	91.6	8.4
تِلْفَازْ	تَلْفِزْيُونْ	12	11	1	91.6	8.4
طَرَابْلُسْ	طْرابْلُسْ	14	14	0	100	0
Total		94	91	3	96.8	3.2

Table 5: Test results for 171 word utterances

Word	Articulation	# of Utterances	Correct	Wrong	%accuracy	%word error rate
أنا	أنا	25	25	0	100	0
سِتَة	سِتِّ	25	25	0	100	0
إثنَين	تَنِينْ	21	18	3	85.7	14.3
سَبْعِينْ	سَبْعِينْ	27	27	0	100	0
شُبَاطْ	شْبَاطْ	18	15	3	83.3	16.7
تِلْفَازْ	تَلْفِزْيُونْ	29	28	1	96.5	3.5
طَرَابْلُسْ	طْرابْلُسْ	26	26	0	100	0
Total		171	164	7	95.9	4.1

The training corpus used in all three tests is composed of 48,500 samples. From this corpus, a 128 phoneme-like templates are produced. In all the tests, the word templates used are composed of three utterances for every word. The seven words used to test the system were not included in the training corpus. In every experiment done, both the training samples and word templates uttered are from the same speaker. In the first test, the DTW is used without applying slope constraint. Here, 158 utterances, at least 20 utterances per word, are used to test the system. The results are shown bellow in Table 3.

The next two tests are done on the ASR system were a slope constraint is applied to the DTW. Both tests showed applying slope constraint gave good improvement in the recognition accuracy rate of the system. The second experiment used 94 utterances for testing, while in the third experiment 171 utterances are used. Tables 4 and 5 show the results of the two testing sets, respectively.

Results of previous work done on Arabic ASR systems are briefly described here in order to check if the recognition results of our system are acceptable. For systems dedicated to standard Arabic, the only large vocabulary system[5,6] had a performance of %15 Word Error Rate (WER). The next system[7], which is a medium sized vocabulary system gave results that ranged from %8 and %4.2 WER. Small vocabulary isolated ASR systems like[9] obtained word error rate that ranged between %15 and %0, while[8] had a % 2.14 WER. Systems that concentrated on the Egyptian colloquial Arabic were all large vocabulary systems. The BBN system had a % 71.1 WER[10], % 61.1 WER[11] and in a more recent version had a %55.8 WER[2]. While in[2] a best performance result of % 53.8 WER was attained, in[12] the result ranged from 55.3 to % 41.7 WER and in[1] the performance ranged from 61 to 39.4% WER.

CONCLUSION

In this study, we have described the development of an automatic speech recognition system based on phoneme-like template model that is dedicated to the Lebanese dialect. The research represented here, to our knowledge, is the first attempt to implement a speech recognition system for the Lebanese colloquial Arabic language. The recognition results produced by our system showed to be satisfactory and when compared, they can match with the results of other Arabic ASR systems. Future work will focus on improving the phoneme-like templates by increasing the size of the training corpus, by trying different distortion measures, calculating 1st and 2nd order delta coefficients and by having multiple codebooks.

REFERENCES

1. Vergyri, D., K. Katrin, D. Kevin and A. Stolcke, 2004. Morphology-based language modeling for Arabic speech recognition. Proc. ICSLP, Jeju, South Korea.
2. Kirchhoff, K., *et al*., 2002. Novel approaches to Arabic speech recognition. Final Report from the JHU Summer Workshop, Tech. Rep., John Hopkins University.
3. Furui, S., 2001. Digital Speech Processing, Synthesis and Recognition. New York, Marcel Dekker, Inc.
4. Huang, X., A. Alec and H.H. Wuen, 2001. Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Upper Saddle River, Prentice Hall.
5. Billa, J. *et al*.,2002. Arabic speech and text in tides ontap. Proc. HLT.
6. Billa, J. *et al*., 2002. Audio indexing of broadcast news. Proc. ICASSP.
7. Lazli, L. and M. Sellami, 2003. Speaker independent isolated speech recognition for arabic language using hybrid HMM-MLP-FCM system. AICCSA, Tunisia.
8. Bahi, H. and M. Sellami, 2004. A connectionist expert approach for speech recognition, The Intl. Arab J. Inform. Technol.
9. El Choubassi, M.M. *et al*., 2003. Arabic speech recognition using recurrent neural networks. IEEE Intl. Symp. Signal Processing and Information Technology ISSPIT, Germany.
10. Billa, J. *et al*., 1997. Multilingual speech recognition: The 1996 BYBLOS CallHome System. Proc. Eurospeech, pp: 363–366.
11. Zavaliagkos, G. *et al*., 1998. The BBN byblos 1997 large vocabulary conversational speech recognition system. Proc. ICASSP.
12. Kirchhoff, K. and D. Vergyri, 2004. Cross-dialectal acoustic data sharing for Arabic speech recognition. Proc. ICASSP. Montreal, Canada.
13. Tarazy, F.H., Al Aswat wa Makharej Al Hrouf Al Arabiet, 1962, Beirut, Matbaet Dar Al Kotob.
14. Nakhle, E.Y., R. Gharaeb and A.A. Alsureit, 1962, Beirut, Al Matbaa AlKatholekiet.
15. Markowitz, J.A., 1996. Using Speech Recognition. Upper Saddle River, Prentice Hall.
16. Becchetti, C. and L.P. Ricotti, 1999. Speech Recognition: Theory and C++ Implementation. Chichester, John Wiley & Sons.
17. Rabiner, L.R. and B.Gold, 1975. Theory and Application of Digital Signal Processing. Englewood Cliffs, Prentice Hall.
18. Sugamura, N., S. Kiyohiro and S. Furui, 1983. Isolated word recognition using phoneme-like templates. Proc. ICASSP, pp: 723-726, Boston, U.S.A.
19. Linde, Y., B. Anres and R.M. Gray, 1980. An algorithm for vector quantizer design. IEEE Trans. Communications, 28: 84-95.
20. Gray, R.M., 1984. Vector Quantization. IEEE

ASAP Mag., 1: 4-29.

21. Jelinek, F., 2001. Statistical Methods for Speech Recognition. Cambridge, The MIT Press.
22. Gray, R.M., B. Andres, G.H. Augustine and Y. Matsuyama, 1980. Distortion measures for speech processing. IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-28: 367-376.
23. Abut, H., R.M. Gray and G. Rebolledo, 1982. Vector quantization of speech and speech-like waveforms. Trans. Acoustics, Speech and Signal Processing, ASSP-30: 423-435.
24. Sakoe, H. and S. Chiba, 1978. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-26: 43-49.
25. Myers, C., R.R. Lawrende and A.E. Rosenberg, 1980. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-28: 623-635.