Speech Recognition using Artificial Neural Networks and Hidden Markov Models

Mohamad Adnan Al-Alaoui[1], Lina Al-Kanj[1], Jimmy Azar[1], and Elias Yaacoub[1]
[1] American University of Beirut/ECE Department, Beirut, Lebanon

***Abstract*—In this paper, we compare two different methods for automatic Arabic speech recognition for isolated words and sentences. Isolated word/sentence recognition was performed using cepstral feature extraction by linear predictive coding, as well as Hidden Markov Models (HMM) for pattern training and classification. We implemented a new pattern classification method, where we used Neural Networks trained using the Al-Alaoui Algorithm. This new method gave comparable results to the already implemented HMM method for the recognition of words, and it has overcome HMM in the recognition of sentences. The speech recognition system implemented is part of the Teaching and Learning Using Information Technology (TLIT) project which would implement a set of reading lessons to assist adult illiterates in developing better reading capabilities.**

***Index Terms*—Al-Alaoui Algorithm, Artificial Neural Networks, cepstral feature extraction, hidden Markov model.**

I. INTRODUCTION

Illiteracy is a serious problem that the Arab world is facing nowadays. In order to solve this problem, the education provided should not only solve illiteracy, but also provide the skills required to react successfully to the rising challenges of technology and the information age.
In this paper, we present part of a new approach to combat language and computer illiteracy at the same time: **Teaching and Learning using Information Technology (TLIT).** With this done, users of this approach will be familiar with reading, writing, and understanding the standard Arabic language, as well as being acquainted with computer and information technology. We will embark upon the speech recognition component of the TLIT project. This component takes the speech produced by the learner and performs recognition in order to determine whether or not the spoken word is the one he/she ought to say. This will serve as a major language teaching tool for the users of the system. It is crucial to note at this point that the environment in which the product will be used is a friendly one, and no adversary to the system is anticipated. That is, the user of the product will be cooperating with the system in an efficient manner, and thus the system's performance would be enhanced. For instance, in case an error in speech recognition was encountered, the system can prompt the user to repeat his task.

The work in this field includes researching different speech recognition algorithms that are the most useful and practical to use in the Arabic language, testing these algorithms, and picking up the best one.

Isolated word/sentence recognition requires the extraction of features from the recorded utterances followed by a training phase. For the feature extraction phase, we have used cepstral coefficients. For the classification phase, we applied neural networks trained using the Al-Alaoui algorithm and compared it against the hidden Markov model classification approach.

Section II presents a summary of the feature extraction method used. Section III describes the hidden Markov model classification method. Section IV describes the new implemented pattern classification method. Section V describes the Arabic speech database used. In Section VI, a comparison between the two methods is performed for isolated words. Section VII presents a comparison between the two classification methods for isolated sentences. Finally, Section VIII presents the graphical user interface that was implemented followed by a conclusion and suggestions for future work.

II. FEATURE EXTRACTION

A. A/D Conversion

The input speech signal is changed into an electrical signal by using a microphone. Before performing A/D conversion, a low pass filter is used to eliminate the aliasing effect during sampling.

A continuous speech signal has a maximum frequency component at about 16 KHz. The sampling rate of the A/D converter should be at least double according to the Nyquist Criterion. Therefore a sampling rate of 32 KHz should be used. The resulting speech samples are now discrete in time but have continuous amplitude values which should be converted to discrete values to be suitable for digital systems. The continuous amplitudes are changed into discrete ones through quantization.

B. Pre-emphasis

Before the digital speech signal can be used for feature extraction, a process called *pre-emphasis* is applied. High frequency formants have lower amplitudes than low frequency formants. Pre-emphasis aims at reducing the high spectral dynamic range. Pre-emphasis is accomplished by passing the signal through an FIR filter whose transfer function is given by: [1]

$$H(z) = 1 - az^{-1} \quad \text{where } 0.9 \le a \le 1 \qquad (1)$$

A typical value for the pre-emphasis parameter 'a' is usually 0.95. [2]

C. *Frame Blocking*

The idea of segmentation of the speech wave into frames, or what is known as frame blocking, comes from the fact that the vocal tract moves mechanically slowly, and as a result, speech can be assumed to be a random process with slowly varying properties [3]. Hence, the speech can be divided into frames, over which the speech signal is assumed to be stationary with constant statistical properties. Another property must be guaranteed to ensure the continuity of the speech signal; this is generally done by overlapping the different frames. The ratio of overlapping must be determined according to the application being developed in order to ensure a high correlation between LPC estimated coefficients of consecutive frames. Note that typical values for the frame period are 45 ms with a 15 ms separation. This corresponds to a 66.7 Hz frame rate.

D. *Windowing*

The process of frame blocking is followed by windowing in order to reduce the energy at the edges and decrease the discontinuities at the edges of each frame, consequently preventing abrupt changes at the endpoints. The mostly used window is the Hamming window, with the following function:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2n\pi}{N-1}\right) \quad 0 \le n \le N-1 \qquad (2)$$

where N is the length of the window. An example is shown in Fig. 1 of a Hamming window of length 101, generated using the MATLAB command wvtool.

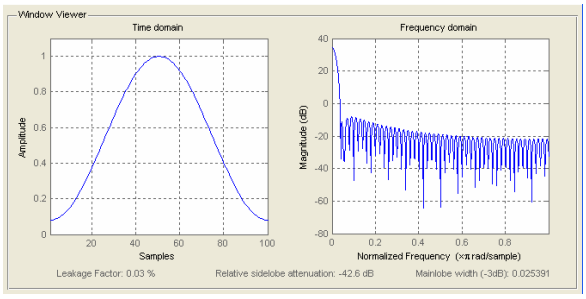


Figure 1. Hamming Window (time domain, frequency domain)

E. *Linear Predicitve Coding*

The Levinson-Durbin algorithm is used to derive the LPC parameter set from an autocorrelation computation with the highest autocorrelation value, p, being the order of the LPC analysis. The algorithm proceeds as shown in Table I.

TABLE I.
THE LEVINSON-DURBIN ALGORITHM

Autocorrelation	$r_l(m) = \sum_{n=0}^{N-1-m} \tilde{x}_l(n)\tilde{x}_l(n+m),\ m = 0,1,\ldots,p$
Initialization	$E^{(0)} = R(0)$
Iterations for $1 \le i \le p$	$k_i = \dfrac{\left\{r(i) - \sum_{j=1}^{L-1} a_j^{(i-1)} r(\lvert i-j\rvert)\right\}}{E^{(i-1)}}$ $\alpha_i^{(i)} = k_i$ $\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$ $E^{(i)} = (1-k_i^2)E^{(i-1)}$

An improvement to the set of LPC coefficients is the derivation of the set of cepstrum coefficients which are the coefficients of the Fourier Transform of the log-magnitude spectrum. Table II presents an iterative algorithm for the determination of the cepstral coefficients.

TABLE II.
CEPSTRUM DETERMINATION

$c_0 = \ln \sigma^2$
$c_m = a_m + \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k},\ 1 \le m \le p$
$c_m = \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k},\ m > p$

σ^2 is the gain term in the model. a_m is an LPC coefficient. We usually choose a set of cepstral coefficients of size Q, where Q is normally taken to be equal to $1.5p$.

Two factors oblige us to modify further the cepstral coefficients. First of all, the low order cepstral coefficients are sensitive to the spectral slope, and the high order cepstral coefficients are sensitive to noise [3]. A solution to these two problems is by weighting these coefficients by a window. A suitable window is a bandpass filter of the form:

$$w_m = \left[1 + \frac{Q}{2}\sin\left(\frac{\pi m}{Q}\right)\right], \quad 1 \le m \le Q \qquad (3)$$

And the weighted coefficients will be:

$$\hat{c}_m = w_m c_m \qquad 1 \le m \le Q \tag{4}$$

F. Vector Quantization

The idea of vector quantization comes from the fact that the output of the LPC analysis is a series of p-dimensional vectors, *p* being the order of the LPC filter. Furthermore, we can argue that it is enough to have one spectral representation for each unique speech unit. For recognition systems that use hidden Markov models, it is important to be able to estimate probability distributions of the computed feature vectors. Because these distributions are defined over a high-dimensional space, it is often easier to start by quantizing each feature vector to one of a relatively small number of template vectors, which together from a codebook.

Vector Quantization usually requires a set of spectral analysis vectors in order to determine the optimal set of codebook vectors. It also requires some measure of similarity between certain vectors. The output of the algorithm will be a group of clusters each having a centroid. The vector space is divided into cells called Voronoi regions. As a result, speech spectral analysis vectors would choose the code book closest to the input vector and the codebook index will be used as the spectral representation. [3]

The essential elements in vector quantization are the distortion measure, the distance measure, and the clustering algorithm. The distance measure represents the distance between the input vector and the codebook vector. The distance measure is given by:

$$d(x, y_i) = \sqrt{\sum_{j=1}^{k} (x_j - y_{ij})^2} \qquad i = 1,.....,N \tag{5}$$

where x_j is the j^{th} component of the input vector x, and y_{ij} is the j^{th} component of the codebook vector y_i.

The average distortion *D* is given by:

$$D = \lim_{M \to \infty} \frac{1}{M} \sum_{n=1}^{M} d(x, y) \tag{6}$$

A quantizer is then said to be optimal if the overall distortion is minimized over all K-level quantizers.

The last very important key element is the Lloyd's algorithm or the K-means clustering algorithm which is used to cluster *M* training vectors into *K* codebook vectors. The creation of the codebook uses an algorithm called the Linde, Buzo, Gray vector quantization algorithm.

III. Hidden Markov Model Representation

The Hidden Markov Model (HMM) is a statistical method that has been used extensively in the field of automatic speech recognition. Given data representing an unknown speech signal, a statistical model of a possible utterance is chosen that most likely resembles the data. Therefore, every possible speech utterance should have a model governing the set of likely acoustic conditions that realize it. The input to an HMM model is an observation vector consisting of a discrete sequence of codebook indices that have been generated by compressing the feature vectors using vector quantization. Comparison against the HMM model of the word uncovers the speaker's utterance. A word is constructed by moving from one state to another according to the state transition probability distribution. Different transitions correspond to variations in the intonations and articulations of the same word.

Since the observation vector consists of discrete codebook indices, the HMM model used in our application is referred to as discrete Markov Model. The use of discrete Markov Model removes the burden of computing continuous probability distributions. HMM probability distributions are summarized in Table III [1].

TABLE III.
MODEL PARAMETERS

Transition Probabilities	$A_{ij} = P\{s_{t+1} = j \mid s_t = i\} \ \forall t$
Observation Probabilities	$B_i(O) = P\{O_t = O \mid s_t = i\} \ \forall t$
Starting Probabilities	$P_i = P\{s_1 = i\}$

AUXILIARY DISTRIBUTION

Forward Variable	$a_t(i) = P\{O_1, O_2, ..., O_t, s_t = i\}$
Backward Variable	$\beta_t(i) = P(O_{t+1}, O_{t+2}, ..., O_T \mid s_t = i\}$
Specific-Time Transition	$\xi_t(i, j) = P\{s_t = i, s_{t+1} = j \mid O_{1...T}\}$
A posteriori Probabilities	$\gamma_t(i) = P\{s_t = j \mid O_{1...T}\}$

Table III defines the state transition probability distribution, ***A,*** which is usually modeled as a matrix $[A_{ij}]$, where transitions occur from state i to state j. It also defines the observation symbol probability distribution, $B = \{B_i(O)\}$ where the probability distribution of each observation is given based on the state that produces it. Note that each observation can be produced by one or more states, and some states may not produce a particular observation at all.

Finally, we also have the initial state distribution $P = \{P_i\}$ which gives the probability that the initial state is state i.

The three probability measures A, B, P are denoted by $\lambda = (A, B, P)$. First order Markov chains are usually used in automatic speech recognition due to their simplicity of implementation.

Two HMM problems arise: training which is the problem of choosing the best HMM model for each word taking into account the variations in utterances, and recognition which is concerned with determining the HMM model that best resembles the utterance. The two problems can be solved using the forward and backward probabilities. The following relations hold [1].:

$$a_{t+1}(j) = B_j(O_{t+1}) \sum_{i=1}^{N} a_t(i) A_{ij} \tag{7}$$

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) x A_{ij} x \mathbf{B}_j(O_{t+1}) \tag{8}$$

Training involves modeling the HMM for each word in the database. This implies that we need to optimize the model parameter $\lambda = (A, B, P)$ to maximize the likelihood of the training set of observation vectors given each word model (i.e. estimating λ that maximizes $P(O|\lambda)$. This problem is crucial since it builds the HMM models during the training phase in a way that guarantees a high probability of correct word recognition later on.

The solution we have applied to the above problem is the **Baum-Welch Algorithm (Forward-Backward Algorithm)**. Baum-Welch is an iterative procedure used to optimize the HMM parameters given the observation vectors for each word. The procedure is repeated iteratively for all the training set. The estimation accuracy is proportional to the amount of training data; therefore a large training set should be used during the training phase to obtain accurate model estimation. This corresponds to recording several utterances of the same word said by different users.

The Baum-Welch algorithm re-estimates the model parameters through ξ and γ using the forward and backward variables [1]:

$$\xi_t(i,j) = \frac{\alpha_t(i) \text{ x } A_{ij} \text{ x } \beta_{t+1}(j) \text{ x } B_j(O_{t+1})}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) \text{ x } A_{ij} \text{ x } \beta_{t+1}(j) \text{ x } B_j(O_{t+1})} \tag{9}$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{10}$$

The parameter models are computed as: [1]

$$\bar{P}i = \gamma_1(i) \qquad \bar{A}ij = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad \bar{B}i(O) = \frac{\sum_{t=1, O_t = O}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{11}$$

Note that at the beginning of the Baum-Welch algorithm, an initial estimation of λ is made. This first estimation must be chosen carefully or else the algorithm might converge to a local minimum and would therefore diverge from the optimal solution. A possible solution to the initial estimation problem is based on the k-means procedure.

IV. AL-ALAOUI NEURAL NETWORKS ALGORITHM

Al-Alaoui algorithm for pattern recognition was originally developed for single layer neural networks [4-8]. It was adapted later to multi layer neural networks [9-12]. The algorithm consists in cloning the erroneously classified samples and adding the resulting clones to the population of the training set to yield a better approximation to the optimum Bayes classifier. Thus it provides better justification and motivation for replicating the erroneously classified patterns during training than the rather ad-hoc boosting approach. The algorithm viability is demonstrated in many practical applications [7-8, 11-12].

NEURAL NETWORKS

The neural network (NN) used in the model was a multilayer perceptron (MLP) with two layers of neurons. The number of neurons in the hidden layer is dependent on the size of the input vector [13]. The output layer has two neurons. The first neuron predicts if the input is a truly spelled word or sentence. The second neuron predicts if the input is a wrongly spelled word or sentence.

The NN is trained to predict one true word or sentence at a time and whichever of these neurons gives the higher score wins. The model is shown in Fig. 2.

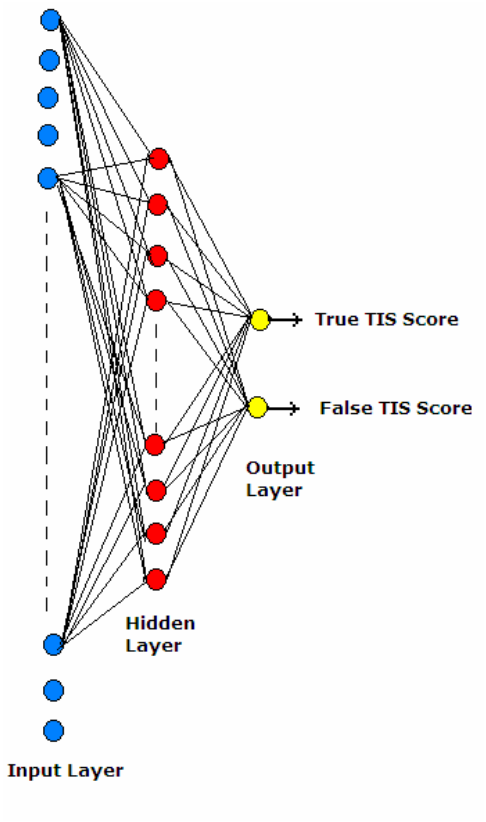


Figure 2. Neural Network

If an MLP network has n input nodes, one hidden-layer of m neurons, and two output neurons, the output of the network is given by

$$y_i = f_i\left(\sum_{k=1}^{m} w_{ki} f_k\left(\sum_{j=1}^{n} w_{kj} x_j\right)\right) \quad (12)$$

where $f_k, k = 1,2,...,m$, and f_i, $i = 1,2$ denote the activation functions of the hidden-layer neurons and the output neurons, respectively; w_{ki} and w_{kj}, $j = 1,2,...,n$ denote the weights connected to the output neurons and to the hidden-layer neurons, respectively; x_j denotes the input. The output activation function was selected to be unipolar sigmoidal:

$$f(u) = \frac{\alpha}{1 + e^{-\beta u}} \quad (13)$$

and the hidden-layer activation functions took the form of hyperbolic tangent sigmoidals for all k:

$$f_k(u) = \alpha_k \frac{e^{\beta_k u} - e^{-\beta_k u}}{e^{\beta_k u} + e^{-\beta_k u}} \quad (14)$$

The weights of the network were learned with the backpropagation method using Al-Alaoui algorithm [5] which iteratively repeats the misclassified samples.

A. *Training*

The generalized inverse algorithm for pattern recognition (backpropagation method using Al-Alaoui algorithm) [5] was used to train the neural network where the method iteratively repeats the misclassified samples in the training. There exist two methods to stop repeating the misclassified samples; either by specifying certain number of iterations in which the misclassified samples are repeated in the training or until there is not a misclassified sample any more. The number of epochs in the training phase differs from one example to another. If the number of epochs is set to be high, the NN will saturate or there will be an over fitting of the NN. This case should be always avoided by setting an acceptable number of epochs. Then, the Al-Alaoui algorithm comes to adapt the NN with the misclassified samples. The momentum is initially set to 0.01 and the learning rate is initially set to 0.1, for each hidden and output neurons, and updated iteratively. The weights and biases of the neural network were initialized in [-1.0, 1.0] using the Nguyen-Widrow method. Desired outputs were set to either 0.9 or 0.1 to represent the true or false site at the output, correspondingly.

V. The Arabic Speech Database

To test our speech recognition algorithm, we had to construct an efficient sample speech database. The database contains a collection of uttered words that comprises Arabic language teaching lessons available in books for the illiterates, supplied to us via the Lebanese Ministry of Education. Each lesson would train the user on pronouncing a number of words. Our software proceeds by asking the user to repeat a certain word (or phrase) printed on the screen and then it prompts him/her if he/she had pronounced it correctly. For testing our algorithm, we chose only one lesson from the given ones, and we recorded the words using a microphone. We recorded the database using our voices, one male and one female. It is important to note that we took into account female and male voice separation, so that when the user of the system is a female, only the female voice will be selected from the database. Similarly, when the learner is a male, only the male utterances of the word in concern will be used. This technique proved to yield better results than performing no gender voice separation.

In order to build the comprehensive database, we will have to record all the words available in all the lessons in the same manner as we recorded the sample database, that is, to record ten or more different utterances of the same word by the same speaker, and performing gender voice separation. But, we will have to allow for the recording of different voices, not only our voices, as this would enhance the algorithm's accuracy in terms of recognition and allow for speaker independence.

The data set for the training and testing phases are recorded by using a PC-based audio input device or a microphone. For simplification purposes the recoded samples are stored in matrices whose rows or columns represent a sample. The recorded signal has values ranging between [-1.0, 1.0] as shown in Fig. 3 for the recorded Arabic word 'مال' spelled in English as 'mal'.

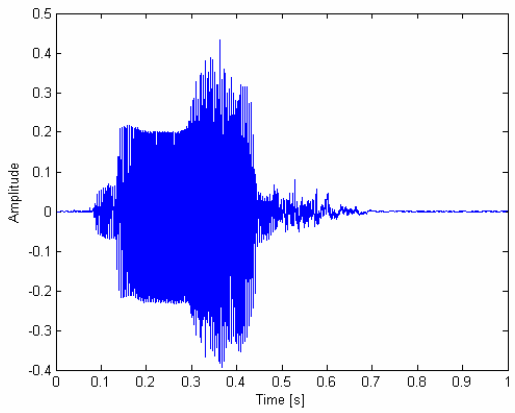


Figure 3. Sample of recorded word 'mal'

For this input signal, the speech signal processing and analysis is done according to Section II, and the output weighted cepstral coefficients are the input to the pattern classifier. The results of training and testing datasets are evaluated using standard performance measures defined as follows:

$$\text{Sensitivity} = \frac{T_P}{T_P + F_N}, \text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$$

$$\text{Specifity} = \frac{T_N}{T_N + F_P}, \text{and} \quad \text{Precision} = \frac{T_P}{T_P + F_P} \qquad (15)$$

where T_P is the number of truly positive samples (words or sentences) predicted as positive, F_P is the number of real negative samples predicted as positive, T_N is the number of real negative sentences predicted as negative and F_N is the number of real positive samples predicted as negative.

VI. Isolated Word Prediction

A. Prediction Using Neural Networks

To train a NN for the prediction of a word, for example "mal", it is more efficient for the rapid convergence of the gradient to train it in parallel with closely spelled words such as "amal" and "malan". The NN should detect only "mal" as the truly detected word so the outputs for the training words "mal" and "malan" are set to 0.9 and 0.1 respectively, whereas the outputs for the words "amal" and "mal" are set to be 0.1 and 0.9 respectively. The number of neurons in the hidden layer of the NN affects significantly the performance of the neural network, and we chose it to be 150 whenever the input vector size of the weighted cepstral coefficients is 3300. In the training phase the number of epochs is set to 40, then the training of the misclassified samples is done using the Al-Alaoui algorithm; for better results the misclassified samples are repeated until there are no more misclassified samples. The number of iterations of the Al-Alaoui algorithm depends on the initial condition of the weights and biases, every time we run a simulation in Matlab these values are assigned randomly. In some cases it takes around 16, 24 or even above 80 iterations. Even if the NN has no more misclassified samples, it may not give good results for the testing sets. In this case another NN should be run and the decision whether it is an acceptable or an unacceptable classifier should be based upon the accuracy of detection of the testing set.

The training set is formed of 10 words 'mal', 10 words 'amal' and 10 words 'malan'. The testing set is formed of 20 words 'mal', 20 words 'amal' and 20 words 'malan'. For a simulation run it took 25 iterations to go out of the Al-Alaoui loop, with all the training set being recognized correctly. For the testing set, the results were not acceptable, since on average 9 of the true 20 words were only predicted as true and 26 of the wrong words were predicted as wrong.

Since the results were not satisfactory, we had two options for increasing the accuracy of the NN: either by increasing the training set or by aligning the beginning of all words. If we look in details at the beginning of the words in the training set we will notice that the words in the recording phase may start at different times as shown in Fig. 4.

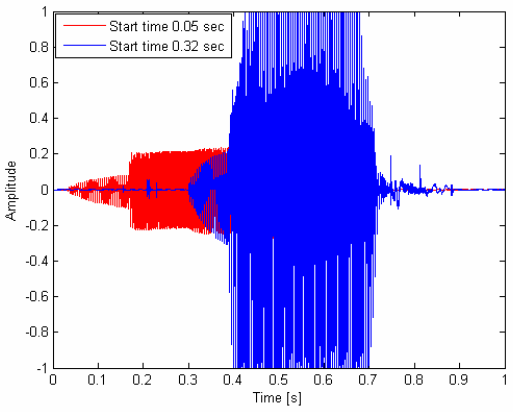


Figure 4. Different starting times in the recording phase

B. Aligning Algorithm

The alignment is done according to the following steps:

1. The word is divided into frames of length 20ms.
2. The energy in each frame is calculated.
3. The first time the energy exceeds the value 0.3 in a frame, the word is considered to start from this frame onward.
4. The length of frames cut from the beginning is appended as zero frames at the end of the word.

An example of a word 'mal' before and after alignment is shown in Fig. 5.

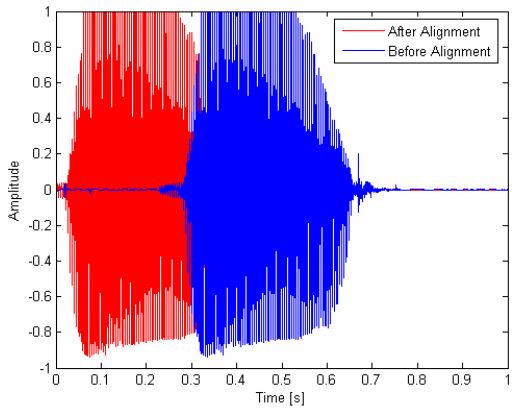


Figure 5. Word 'mal' before and after alignment.

Now, if we look at a set of aligned words, we will notice much more coherence in the behavior as shown in Fig. 6, where 5 aligned words of 'mal' are shown.

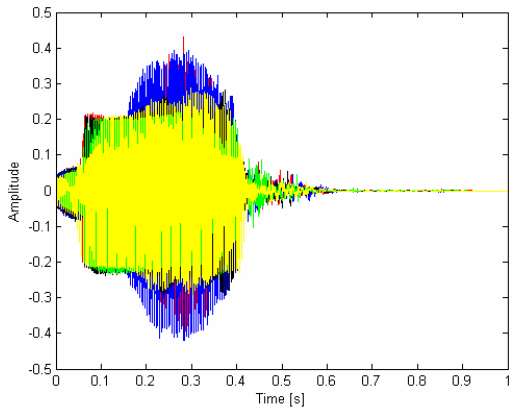


Figure 6. Coherent behavior of five words 'mal' plot together

Now, if we train the NN with the previous training set but with the words aligned, we obtained a NN that passed in the Al-Alaoui loop for 17 iterations to give 100% accurate detection of the training set, and gave around 99% accurate detection for the testing set, where 19 of 20 true were predicted as true and 40 of 40 false were predicted as false.

C. *Comparison with HMM*

The same training set was used to train an HMM to predict the word 'mal'. Table IV shows the results that compare the NN to the HMM.

TABLE IV.
COMPARISON BETWEEN HMM & NN FOR WORD PREDICTION

	NN	HMM
Sensitivity (Training)	10/10*100 = 100%	4/10*100 = 40%
Specificity (Training)	20/20*100 = 100%	--------
Accuracy (Training)	30/30*100 = 100%	4/10*100 = 40%
Sensitivity (Testing)	19/20*100 = 95%	5/20*100 = 25%
Specificity (Testing)	40/40*100 = 100%	12/40*100 = 30%
Accuracy (Testing)	59/60*100 = **98.33%**	17/60*100 = **28.33%**

It is clear that the NN overcomes HMM with such a small number of samples in the training set. The HMM is trained with the words that should be truly detected and thus as opposed to the NN it is not trained in parallel with the words that should be falsely detected. The HMM can be improved by increasing the samples of the training set, but this is a disadvantage compared to the NN, and especially, if the pattern classifier has to classify the words truly said by more than one speaker.

VII. ISOLATED SENTENCE PREDICTION

Our first attempt was to perform speech recognition of sentences through speech segmentation.

A. *Speech Segmentation Using Average Level Crossing Rate Information*

The level crossing rate (LCR) at a sample point is defined as "the total of all the level crossings that have occurred for that level over a short interval around the point divided by the interval duration"[14]. The average level crossing rate (ALCR) at each point is the summation of the level crossing rate over all levels. The ALCR-based Segmentation Algorithm is described in [14].

The result of segmentation is shown in Fig. 7. Black crosses designate the start of a segment and red crosses designate the end of a segment. The black line shows the ALCR curve.

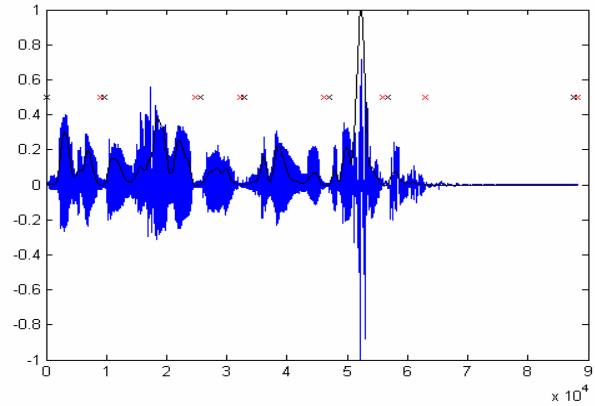


Figure 7. Segments obtained using ALCR method

We placed an extra condition on the resulting segments, mainly that segments that are less than 0.3sec in duration are included into the previous segment. The result is shown in the Fig. 8.

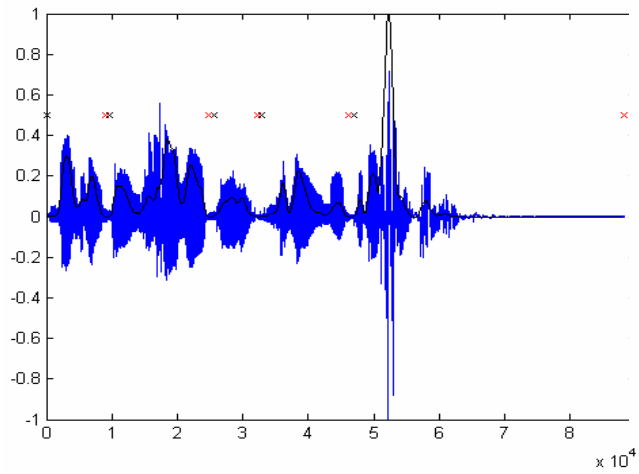


Figure 8. Adding condition to ALCR method

The segmentation algorithm does not yield word segmentation but rather phoneme-based segmentation (despite the extra conditions we have placed into combining different adjacent segments). For example we applied the segmentation algorithm to the Arabic sentence "ذهب الولد إلى المدرسة", written in English as "zahaba alwalado ila almadrasa". Ten sentences were used for testing. The segmentation differed from a sentence to another as shown in Fig. 9, Fig. 10 and Fig. 11 where the same spelled sentences were segmented into 7, 8 and 9 segments respectively as shown by the red segments. Consistent word segmentation was not possible.

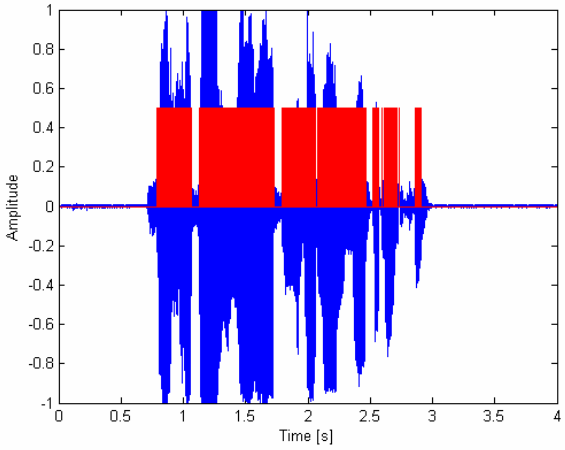


Figure 9. The sentence "ذهب الولد إلى المدرسة" segmented into 7 parts

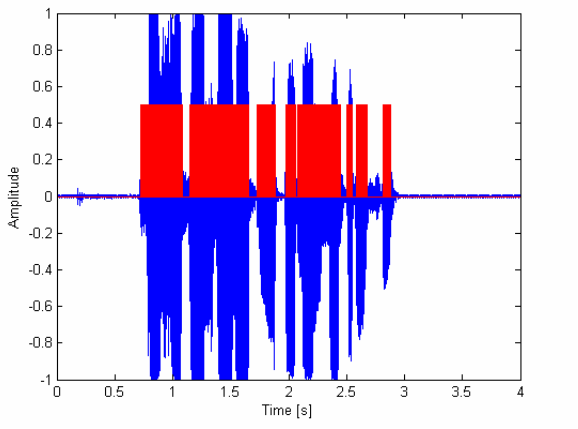


Figure 10. The sentence "ذهب الولد إلى المدرسة" segmented into 8 parts.

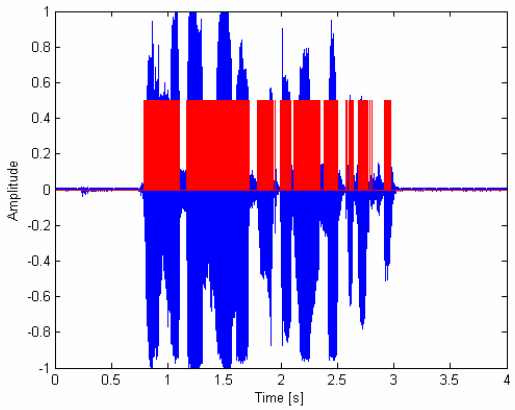


Figure 11. The sentence "ذهب الولد إلى المدرسة" segmented into 9 parts.

This means that segmentation algorithms are not compatible with our word-based HMM system. Furthermore, the variability of the segmentation result is sufficiently high to hinder its use as input to our ANN system. Hence, we shall next experiment with using the weighted cepstrum vectors of full sentences as input to our ANN system.

B. Prediction of relatively different sentences

As a first approach for sentence prediction, we tried to examine a classifier that differentiates between significantly different sentences such as 'أنا أتكلم اللغة العربية' spelled in English as 'Ana atakalam allougha al 3arabiah', and 'أنا من لبنان', spelled in English as 'Ana min lubnan'. The data sets are registered over 3 seconds. First, the data set consisted of 30 sentences for training and 60 sentences for testing. Twenty out of the 30 training sentences were 'أنا أتكلمُ اللغة العربية', and the remaining 10 were 'أنا من لبنان'. The testing set consisted of 40 sentences 'أنا أتكلمُ اللغة العربية' and 20 sentences 'أنا من لبنان'. Here too, the alignment procedure described in Section VI is applied first.

The same steps used for the prediction of a word are applied for sentence prediction. The NN is trained to predict the Arabic sentence 'أنا أتكلم اللغة العربية'. An NN

was trained, with the number of epochs set to 60, and the results shown in Table V (in comparison with HMM) were very satisfactory for such different sentences.

TABLE V.
COMPARISON BETWEEN HMM & NN FOR RELATIVELY DIFFERENT SENTENCE PREDICTION

	NN	HMM
Sensitivity (Training)	20/20*100 = 100%	16/20*100 = 80%
Specificity (Training)	10/10*100 = 100%	--------
Accuracy (Training)	30/30*100 = 100%	16/20*100 = 80%
Sensitivity (Testing)	40/40*100 = 100%	28/40*100 = 70%
Specificity (Testing)	20/20*100 = 100%	19/20*100 = 98%
Accuracy (Testing)	60/60*100 = **100%**	47/60*100 = **78.33%**

Similar to the case of word prediction, the NN overcomes the HMM in the training and testing accuracy.

Though the NN gives excellent results for the testing set, the problem is that it does not differentiate between very close sentences such as 'أنا لا أتكلمُ اللغة العربية' (ana atakallam allougha al arabiya) or 'أنا أتكلمُ اللغة الإنكليزية'(ana atakallam allougha al ingliziya) that differ by only one word from each other.

C. *Prediction of relatively close sentences*

In this part, we will examine the classifier for the prediction of the intended target sentence ' أنا أتكلمُ اللغة العربية' from relatively close sentences. The close sentences are for example 'أنا أتكلمُ اللغة الإنكليزية' and ' أنا لا أتكلمُ اللغة العربية' which differ by only one word from the target sentence. A necessary condition was to increase the training set with relatively close sentences included. First, the data set consisted of 150 sentences for training and 50 sentences for testing. A hundred out of the 150 training sentences were the target sentence 'أنا أتكلمُ اللغة العربية', and the remaining 50 were the close sentences ' أنا أتكلمُ اللغة الإنكليزية', 'أنا لا أتكلمُ اللغة العربية'. The testing set consisted of 50 sentences of each 'أنا أتكلمُ اللغة العربية', ' أنا أتكلمُ اللغة الإنكليزية', 'أنا لا أتكلمُ اللغة العربية' 'أنا من لبنان', and silence.

As our NN method using Al-Alaoui algorithm proved itself superior in sentence prediction, we will proceed with applying the method for close sentence prediction and compare it against the K-Nearest Neighbor classifier. A neural network was trained with the described training set; the number of epochs was set to 100 in this case. The results are shown in Table VI in comparison with K-Nearest Neighbor Classifier. Effectively, an object is classified according to the class corresponding to the largest number of objects of that class in the k-nearest neighbor of the object in question. We used the KNN classifier with k = 1. The train set contained 150 objects, 50 objects within each of 3 classes. These are the weighted cepstral vectors. The test set contained 50 objects, 10 objects within each class.

TABLE VI.
COMPARISON BETWEEN NN & KNN FOR RELATIVELY CLOSE SENTENCE PREDICTION

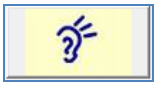
	NN	KNN
Sensitivity (Training)	100/100*100 = 100%	100/100*100= 100%
Specificity (Training)	50/50*100 = 100%	50/50*100 =100%
Accuracy (Training)	150/150*100 = 100%	150/150*100 =100%
Sensitivity (Testing)	4/10*100 = 40%	8/10*100 =80%
Specificity (Testing)	40/40*100 = 100%	40/40*100 = 100%
Accuracy (Testing)	44/50*100 = **88%**	48/50*100 =**96%**

The overall validation error for NN is 6/50 = 12% -- all being false negatives which is acceptable.
The performance of the K-Nearest Neighbor Classifier (with k=1) in this regard was superior to NN yielding a validation error of only 4%.

VIII. GRAPHICAL USER INTERFACE [15]

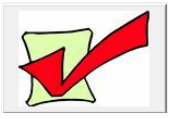
The interfaces developed allow the user to choose whether he is a male or a female since our application is speaker dependent, to go through the different words in the lessons, and to hear the words and the different expressions. The user may also try to read the words and finally, use the speech recognition routine to make sure that he or she read the word correctly.

The major components of the GUI are:



This button allows the user to hear either the word whose image is on the screen or to listen to the different instructions. In fact, since the application will be used for the teaching of illiterates, it is important to simplify the task of the users as much as possible.

This button allows the user to record the word that is shown on the screen for recognition purposes.



This is the most important button. When the button is pressed, the word entered by the user and which was

recorded, will be processed, quantized, then compared with the word model, in order to know if it was recognized correctly or not. If the word is recognized correctly, a recorded message will inform the user that he/she said the word correctly. Otherwise, another recorded message will inform the user that he/she said the word incorrectly advising him/her of repeating the word again.

These buttons are just used to go through the lesson. The final GUI looks as follows:

Figure 12. Choice of gender by the speaker

Figure 13. The sentence 'أنا أتكلمُ اللغة العربية'

IX. Conclusion and Future Work

The aim of the paper was to develop a speech recognition component for TLIT project which would implement a set of reading lessons to assist adult illiterates in developing better reading capabilities. The first stage involved the identification of the different alternatives for the different components of a speech recognition system, such as using linear predictive coding, using Hidden Markov Models, Neural Networks or K-Nearest Neighbor Classifier for the pattern recognition block. The NN classifier trained using the Al-Alaoui Algorithm overcomes the HMM in the prediction of both words and sentences. We also examined the KNN classifier which gave better results than the NN in the prediction of sentences. The segmentation of Arabic sentences was also considered in our work, and we proved the problems in applying it to Arabic speech. In this project we implemented several classifiers for the Arabic speech recognition problem. Many other alternatives for the different components of the system could be considered. For instance, the number of the required training speakers should be determined in order to get a classifier that is able to perform true speech recognition for all speakers. One could use other than LPC analysis for representing the features of the speech signal, and also add a speaker recognition component to the speech recognition component in order to identify the user of the application and as a result, track his progress in the learning process. Whenever these steps will be accomplished and implemented, the system will be developed to become commercial.

Acknowledgements

The authors greatly appreciate the support and cooperation of Ms. Amal Charara of the Lebanese National Committee for Literacy at the Lebanese Ministry of Social Affairs, and Ms. Nour Dajani of the UNESCO. Gratitude also goes to George Kadifa, President and CEO of Coreo Inc. and to Ibrahim Hajj, Dean of the Faculty of Engineering and Architecture at the American University of Beirut, for their support and encouragement. This research was supported by grants from the Rathmann Family Foundation, UNESCO, and URB (University Research Board) of AUB.

References

[1] Naous, Y. S., G. F. Choueiter, M. I. Ohanessian, and M. A. Al-Alaoui. 2002. "Speech Recognition for Voice-Based Control", *Proceedings of the 2nd IEEE International Symposium on Signal Processing and Information Technology*, IEEE ISSPIT 2002, 527–531. Marrakesh, Morocco, December 18–21, 2002.

[2] Becchetti C., Lucio R., "Speech Recognition, Theory and C++ Implementation". John Wisely and Sons Inc., England, 1999.

[3] Rabiner L., Juang B., "Fundamentals of Speech Recognition". PTR Prentice Hall, NJ, 1993.

[4] M. A. Al-Alaoui, " Some Applications of Generalized Inverse to Pattern Recognition," Ph.D. Thesis, Electrical Engineering Department, Georgia Institute of Technology, December, 1974.

[5] Al-Alaoui, M. A. "Applications of Constrained Generalized Inverse to Pattern Classifications"; Pattern Recognition, Vol. 8, No. 4, pp. 277-281, 1976.

[6] M. A. Al-Alaoui, " A New Weighted Generalized Inverse Algorithm for Pattern Recognition," IEEE Transactions on Computers, Vol. C-26, No. 10, pp. 1009-1017, October 1977.

[7] M. A. Al-Alaoui, J. El Achkar., M. Hijazi, T. Zeineddine, and M. Khuri, "Application of Artificial Neural Networks to QRS Detection and LVH Diagnosis"; Proceedings of ICECS'95, pp. 377-380, Amman-Jordan, 17-21 December 1995.

[8] R. Wouhaybi , and M. A. Al-Alaoui , "Comparison of Neural Networks for Speaker Recognition", Proceedings of The Sixth IEEE International Conference on Electronics, Circuits and Systems (ICECS'99), Pafos, Cyprus, Vol.1, pp. 125-128, September 5 - 8, 1999.

[9] M. A. Al-Alaoui, R. Mouci, and M. Mansour, M. "A redundancy approach to classifier training" , Proceedings of the 7th IEEE International Conference on Electronics Circuits and Systems (ICECS'2K), Vol. II, pp. 950-953, September 17-20, 2000, Jounieh, Lebanon.

[10] Al-Alaoui M.A., R. Mouci, M. M. Mansour, and R. Ferzli, "Cloning Approach to Classifier Training", IEEE transaction on Systems, Man and Cybernetics , Part A, Volume: 32 Issue: 6 , pp. 746-752, Nov. 2002.

[11] J. Azar, H. Abou Saleh, and M. A. Al-Alaoui, "Sound Visualization for the Hearing Impaired," International Journal of Emerging Technologies in Learning" - iJET. March, 2007, pp. 1-7.

[12] Ferzli R. and M. A. Al-Alaoui, "Subsampling Image Compression Using Al-Alaoui Backpropagation Algorithm", the 14th IEEE International Conference on Electronics, Circuits and Systems , Marrakech, Morocco, December 11-14, 2007.

[13] A. Pinkus, "Approximation theory of the MLP model in neural networks", tech. rep., 1999.

[14] A. Sarkar and T.V. Sreenivas, "Automatic Speech Segmentation Using Average Level Crossing Rate Information".

[15] S. Al Hattab, Y. Yaacoub, A. El hajj, "Teaching and Learning Using Information Technology Arabic Speech Recognition Team", FYP, ECE, AUB, May 2007, unpublished.

Manuscript received 09 October 2007.