International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:12, 2007

Speaker Independent Quranic Recognizer Based on Maximum Likelihood Linear Regression

Ehab Mourtaga, Ahmad Sharieh, and Mousa Abdallah

***Abstract*—**An automatic speech recognition system for the formal Arabic language is needed. The Quran is the most formal spoken book in Arabic, it is spoken all over the world. In this research, an automatic speech recognizer for Quranic based speaker-independent was developed and tested. The system was developed based on the tri-phone Hidden Markov Model and Maximum Likelihood Linear Regression (MLLR). The MLLR computes a set of transformations which reduces the mismatch between an initial model set and the adaptation data. It uses the regression class tree, as well as, estimates a set of linear transformations for the mean and variance parameters of a Gaussian mixture HMM system. The 30th Chapter of the Quran, with five of the most famous readers of the Quran, was used for the training and testing of the data. The chapter includes about 2000 distinct words. The advantages of using the Quranic verses as the database in this developed recognizer are the uniqueness of the words and the high level of orderliness between verses. The level of accuracy from the tested data ranged 68 to 85%.

***Keywords*—**Hidden Markov Model (HMM), Maximum Likelihood Linear Regression (MLLR), Quran, Regression Class Tree, Speech Recognition, Speaker-independent.

I. Introduction

PRODUCING high performance speaker independent acoustic models for a large vocabulary recognition system requires huge amounts of data for different speakers. However it is possible to build improved acoustic models by tailoring a model set to a specific speaker. By collecting data from a speaker and training a model set on this speaker's data alone, the speaker's characteristics can be modeled more accurately. Such systems are commonly known as *speaker dependent* systems. On a typical word recognition task, they systems may have half the errors of a *speaker independent* system. The drawback of speaker dependent systems is that a large amount of data (typically hours) must be collected in order to obtain sufficient amount of model accuracy [10].

The goal of the Automatic Speech Recognition (ASR) systems is to transcribe human speech into text, which can be further processed by machines or displayed for humans reading in various applications [7, 8]. Depending on the application, different kinds of recognizers are used. Isolated word recognition is sufficient for simple user interface systems. A harder task is to transcribe continuous speech. Continuous speech recognition is needed, for such things as, aids meant for the hearing or visually impaired.

Rather than training speaker dependent models, *adaptation* techniques can be applied. In this case, by using only a small amount of data from a new speaker, a good speaker independent system model set can be adapted to better fit the characteristics of this new speaker [10]. Speaker adaptation techniques can be used in various different modes. If the true transcription of the adaptation data is known then it is termed *supervised adaptation*, whereas if the adaptation data is unlabelled then it is termed *unsupervised adaptation*. In the case where all the adaptation data is available in one block, e.g. from a speaker enrollment session-then it is termed *static adaptation*. Alternative adaptation can proceed incrementally as adaptation data becomes available, and this is termed *incremental adaptation*.

Adaptation is performed offline by using the Maximum Likelihood Linear regression (MLLR) technique to estimate a series of transforms or a transformed model set, which reduces the mismatch between the current model set and the adaptation data.

To effectively use a speech recognition system, the speech signal has to be parameterized in some appropriate manner. This is often referred to as the signal-processing front end [9]. The parameterization aims to represent the speech signal as compactly as possible-thus reducing the data rate. Many different forms of speech parameterization have been considered over the years. There are two main approaches: one is some type of coding—usually linear prediction of the time domain signal; the other is a direct sampling of domains other than time domain, usually the frequency or cepstral domains. These speech signal analysis approaches are usually referred to as Linear Predictive Coding (LPC) and filterbank analysis or the *Mel-Frequency Cepstral Coefficients* (MFCCs), respectively. The MFCC is the best known and most popular [4], so it will be used here.

The ASR has been investigated from the early 1950s [12]. During the 1980s, statistical modeling-namely the Hidden Markov models (HMMs)-started to replace the earlier template-matching-based methods in ASR [5, 19]. Today, the most successful ASR systems are based on these statistical models, of course flavored with lots of adjustments and improvements. Speech recognizers based upon HMMs have achieved a high level of performance in controlled environments [3,11,18 and 22]. The trend of imitating the

A. Sharieh is with the Computer Science Dept., The University of Jordan, Amman, Jordan (e-mail: sharieh@ju.edu.jo).

E. Mourtaga graduated from Department of Computer Science, The University of Jordan 2005.

M. Abdallah is with Princesses Sumia University, Amman, Jordan.

human vocal tract using mechanical means continued into the 20th century. Gradually, when new algorithms, electronics and finally computers with increasing computational power were developed, the scope of speech research broadened to speech coding and recognition applications.

An early but serious application of speech recognition was the single isolated speaker digit recognizer designed at Bell Laboratories in 1952 [7, 8]. By the mid-1960's, most researchers realized speech recognition was far more subtle and intricate than they had anticipated. In 1975, Jelinek et al. presented the idea of language modeling in the form it is used today [14]. The most fundamental change in the 1980's in the field of speech recognition was the shift from the spectral pattern matching techniques to statistical modeling methods- for example neural networks and more importantly HMMs. Research to improve statistical processing continued into the 1990's until now; and was accompanied by a growing emphasis on developing intelligent, *spoken understanding* systems. The complex human factors issues related to speech recognition began to unfold, moving the industry towards better human factors design.

The Arabic language is one of the most widely used languages around the world with approximately 300 million speakers [2]. One example of working on isolated Arabic word recognition was started at the Faculty of Engineering, Cairo University [1]. The main objective of the work is to build a reliable isolated word recognition system for a limited number of words for a specific purpose such as numeric data entry. In 1999, an Arabic speech recognition model using Artificial Neural Networks (ANN) was developed [15]. At the front-end, the model used the LPC approach. At the back-end, the model used an ANN approach. The model was implemented twice. In the first implementation, the Multi-Layer Feed Forward Neural networks model was used at the back-end. In the second implementation, the Self-Organizing Map with Learning Vector Quantization Neural Networks was used at the back-end. The overall accuracy was reported to be 89.99% for the first technique, and 91.7% for the second technique [15]. The model was limited to a selected set of 73 isolated Arabic words.

Two nonlinear models have been investigated based on the use of back propagation neural networks [2]. This classifier can distinguish between the three Arabic vowels for either male or female speakers. By May 2002, a large vocabulary speech recognition model for Arabic was implemented at the University of Jordan [1]. This model used a hybrid of Hidden Markov Model with Artificial Neural Networks. The tri-phone was used as the smallest unit of recognition to build a dictionary for Arabic. The accuracy of the system when tested using the same training set was 70.88%. Nevertheless, the overall accuracy was 57.5%.

The Quran is the most popular formal book in Arabic. Mourtaga and others presented Quranic based speaker-dependent using the tri-phone/HMM model with the accuracy going up to 80% [17]. This paper presents a work which builds a Quranic speaker-independent recognizer based on the MLLR.

Section 2 explains a model adaptation using MLLR based on HMM. Section 3 presents the proposed recognizer. Section 4 discusses the experiments and the results. Section 5 presents the conclusion and the future research.

II. Model Adaptation Using MLLR

A Hidden Markov Model Adaptation

The HMM is a statistical model that can be used to model time series data. The statistical formulation of the problem can be written as in (1), where $\hat{W}$ is the recognition hypothesis, i.e. the chosen vocabulary item. The HMMs provide efficient means to compute (1). According to the Bayes' formula, (1) can be written as in (2). Since the acoustic information *O* is a fixed variable in (2), *P(O)* does not have influence on the maximization in (2) which therefore can be represented by (3).

$$\hat{W} = \arg\max_{W} P(W|O), \tag{1}$$

$$\hat{W} = \arg\max_{W} \frac{P(W)P(O|W)}{P(O)} \tag{2}$$

$$\hat{W} = \arg\max_{W} P(W)P(O|W) \tag{3}$$

The application of HMMs in speech recognition is based on two assumptions of the speech signal [19]: 1) Speech signal is piecewise stationary, i.e. it can be segmented in such a way that the stochastic characteristics of the signal do not change during each segment, and 2) The adjacent samples of the process- i.e. adjacent feature vectors-are independent of each other. This suggests that no inter-frame correlation exists.

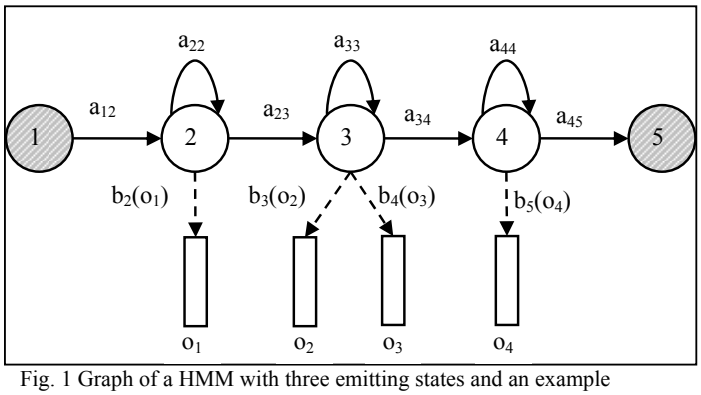

Fig. 1 Graph of a HMM with three emitting states and an example feature vector sequence of this generative model

Fig. 1 shows an example of a HMM with three emitting states. Each state of 2 to 4 is associated with an emission Probability Density Function (PDF) *bj*(***o***). The first and last states of the model are non-emitting. The probability of the current state of the model at a particular time instant depends only on the state at the preceding time instant. The HMMs consist of a Markov chain and state-dependent emission

PDFs. The equations and the related models of HMM are discussed in details in [21].

B. Clustering Mechanism for Context-Dependent HMM's

One way of reducing the number of parameters in a triphone model set is to tie the middle states of the models in the system. The assumption that the center of each triphone is similar, could lead to this kind of approach. However, clustering mechanisms lead to better results than this kind of direct tying. There are two famous approaches for clustering: a data-driven clustering approach and a decision tree-based approach. A limitation with data-driven approach is its inability to deal with unseen triphones (triphones not present in the training data), which are bound to occur in large vocabulary recognition system. So, the decision-tree based system used in this work provides the quality of clustering as offers a solution to the unseen triphone problem [20].

Initially all states in a given list are placed in the root node of a tree. The nodes are iteratively split by selecting a question. Depending on the answer, the states/models in the state are placed either in the right or the left child node of the current node. This is done iteratively until the log likelihood increase of the states/model in the tree node obtained by the best question is below a predefined limit. At this point, all the parameters in the state/model are tied. The log likelihood can be calculated based on the statistics (means, variances, and state occupation counts) gathered from the training data. Based on the previous information the best criteria for each node can be chosen. The models for unseen triphones can be formed by using the trees generated in the clustering processes. You can simply follow the questions from the root of the tree. Once a leaf node is reached, that state/model is used for the unseen triphone in the question.

The MLLR computes a set of transformations that will reduce the mismatch between an initial model set and the adaptation data. It is a model adaptation technique that estimates a set of linear transformations for the mean and variance parameters of a Gaussian mixture HMM system. The effect of these transformations is to shift the component means and alter the variances in the initial system so that each state in the HMM system is more likely to generate the adaptation data. The transformation matrix used to give a new estimate of the adapted mean is given by (4).

$$\hat{\mu} = \mathbf{W}\xi, \tag{4}$$

The $\mathbf{W}$ is the n $\times$(n+1) transformation matrix (where n is the dimensionality of the data) and ξ is the extended mean vector defined in (5). The w represents a bias offset whose value is fixed (within HTK) at 1. The $\mathbf{W}$ can be decomposed as in (6). The A represents an n by n transformation matrix and b represents a bias vector. The transformation matrix $\mathbf{W}$ is obtained by solving a maximization problem using the *Expectation-Maximization* (EM) technique. This technique is also used to compute the variance transformation matrix.

$$\xi = \left[w\mu_1\mu_2 \ldots \mu_n\right]^T \tag{5}$$

$$\mathbf{W} = [\mathrm{b}\ \mathrm{A}] \tag{6}$$

This adaptation method can be applied in a very flexible manner, depending on the amount of adaptation data that is available. If a small amount of data is available then a *global* adaptation transform can be generated. A global transform (as its name suggests) is applied to every Gaussian component in the model set. As more adaptation data becomes available, improved adaptation is possible by increasing the number of transformations. Each transformation is applied to certain groups of Gaussian components. The Gaussian components could be grouped into the broad phone classes: silence, vowels, stops, glides, nasals, fricatives, etc. The adaptation data could be used to construct more specific broad class transforms to apply to these groups.

The MLLR makes use of a *regression class tree* to group the Gaussians in the model set, so that the set of transformations to be estimated can be chosen according to the amount and type of adaptation data that is available. The tying of each transformation across a number of mixture components makes it possible to adapt distributions for which there weren't any observation. With this process, all models can be adapted. The adaptation process is dynamically refined when more adaptation data becomes available. The regression class tree is constructed so as to cluster together components that are close in acoustic space-so that similar components can be transformed in a similar way. The tree is built using the original speaker independent model set, and is thus independent of any new speaker. The tree is constructed with a centroid splitting algorithm, which uses a Euclidean distance measure. For more details see [26]. The terminal nodes of the tree specify the final component groupings, and are termed the *base (regression) classes*. Each Gaussian component of a model set belongs to one particular base class.

C. Estimation of the Mean Transformation Matrix

To enable robust transformations to be trained, the transform matrices are tied across a number of Gaussians. The set of Gaussians which share a transform is referred to as a regression class.

For a particular transform case W_m, the R Gaussian components $\{m_1, m_2 \ldots, m_R\}$ will be tied together, as determined by the regression class tree. By formulating the standard auxiliary function; maximizing it with respect to the transformed mean; and considering only these tied Gaussian components, the following is obtained-as in (7) and $L_{m_r}(t)$, the occupation likelihood, is defined by (13).

$$\sum_{t=1}^{T}\sum_{r=1}^{R}(L_{m_r}(t)\sum\nolimits_{m_r}^{-1}\mathbf{o}(t)\xi_{m_r}^{T}) = \sum_{t=1}^{T}\sum_{r=1}^{R}(L_{m_r}(t)\sum\nolimits_{m_r}^{-1}\mathbf{W}_m\xi_{m_r}\xi_{m_r}^{T}) \tag{7}$$

$$L_{m_r}(t) = P(q_{m_r}(t)\left|M,\mathbf{O}_T\right.) \tag{8}$$

The $q_{m_r}(t)$ indicates the Gaussian component m_r at time t, and $\mathbf{O}_T = \{\mathbf{o}(1),\ldots,\mathbf{o}(T)\}$ is the adaptation data. The occupation likelihood is obtained from the forward-backward process.

To solve for $\mathbf{W}_m$, two new terms are defined:

1. The left hand side of (7) is independent of the transformation matrix and is referred to as Z, and is defined in (9).

$$Z = \sum_{t=1}^{T}\sum_{r=1}^{R}(L_{m_r}(t)\sum\nolimits_{m_r}^{-1}\mathbf{o}(t)\xi_{m_r}^{T}) \quad (9)$$

2. A new variable $\mathbf{G}^{(i)}$ is defined with elements in (10).

$$g_{jq}^{(i)} = \sum_{r=1}^{R} v_{ii}^{(r)} d_{jq}^{(r)} \quad (10)$$

where

$$\mathbf{V}^{(r)} = L_{m_r}(t)\sum\nolimits_{m_r}^{-1} \quad (11)$$

and

$$\mathbf{D}^{(r)} = \xi_{m_r}\xi_{m_r}^{T} \quad (12)$$

It can be seen that from these two new terms, $\mathbf{W}_m$ can be calculated from (13). The w_i is the i^{th} vector of W_m and z_i is the i^{th} vector of Z.

$$\mathbf{w}_i^{T} = \mathbf{G}_i^{-1} z_i^{T} \quad (13)$$

The regression class tree is used to generate the classes dynamically, so it is not known a-priori which regression classes will be used to estimate the transform. This does not present a problem, since $G^{(i)}$ and Z for the chosen regression class may be obtained from its child classes (as defined by the tree). If the parent node R has children $\{R_1,\ldots,R_C\}$, then we have equations (14) and (15). From this it is clear that it is only necessary to calculate $\mathbf{G}^{(i)}$ and $\mathbf{Z}$ for the most specific regression classes possible- i.e. the base classes.

$$\mathbf{Z} = \sum_{c=1}^{C} \mathbf{Z}^{(R_c)} \quad (14)$$

$$\mathbf{G}^{(i)} = \sum_{c=1}^{C} \mathbf{G}^{(iR_c)} \quad (15)$$

D. Clustering Mechanism for Context-Dependent HMM's

Estimation of the variance transformation matrices is only available for diagonal covariance Gaussian systems. The Gaussian covariance is transformed using (16) and (17). The $\mathbf{H}_m$ is the linear transformation to be estimated and $\mathbf{B}_m$ is the inverse of $\sum_m^{-1}$, so we have (16), (17), and (18). After rewriting the auxiliary function, the transform matrix $\mathbf{H}_m$ is estimated by (19).

$$\hat{\sum}_m = \mathbf{B}_m^{T}\mathbf{H}_m\mathbf{B}_m \quad (16)$$

$$\sum\nolimits_m^{-1} = \mathbf{C}_m\mathbf{C}_m^{T} \quad (17)$$

$$\mathbf{B}_m = \mathbf{C}_m^{-1} \quad (18)$$

$$\mathbf{H}_m = \frac{\sum_{r=1}^{R_c}\mathbf{C}_{m_r}^{T}\left[\mathbf{L}_{m_r}(t)(\mathbf{o}(t)-\mu_{m_r})(\mathbf{o}(t)-\mu_{m_r})^{T}\right]\mathbf{C}_{m_r}}{\mathbf{L}_{m_r}(t)} \quad (19)$$

III. The Recognizer

The first step in the HMM training is to define a prototype model. The parameters of this model are used to define the model topology. This paper uses the thirteen states left-to-right model. Then a new version of the prototype is created with the global speech means and variances. The flat start monophones are re-estimated using the embedded re-estimation tool, HERest [6, 22].

Speech utterances may contain pauses of different lengths and nature. There are longer periods of silence at sentence borders and shorter periods inside a sentence. Between most words, the pause is very short or even nonexistent. In stop constants, there is a short silent section, and sometimes there are glottal stops in different locations of speech. Additionally, some background noise is present and should be handled by some models. Therefore, several kinds of silence models are needed. In this paper, two different silence models are used. For the silence model, transitions from the initial to final state and back are added to allow the different states to absorb the background noise without creating too many consecutive (sil) observations in the recognition results. A short pause model (sp) is created. It has only one emitting state, which is tied (set equal) to the center state of the (sil) model. There is also a transition from the beginning state directly to the end state. Another two passes of HERest are applied using the phone transcriptions with sp models between words.

In theory, triphone models are clones of monophone models that have been renamed to include the left and right context of the phoneme and retrained with the occurrences of the triphone in question. Training a full triphone is not feasible due to the large number of possible triphones and the small number of training examples for many of these in any realistic data set. Therefore, the triphones need to be clustered.

Initially, the monophones label files need to be transformed into triphone form. The short pause is ignored when forming triphones. The (sil) label is considered as a normal context to other phonemes. The next step is to make clones of the phoneme models for each triphone in the training data. This is done by the script driven HMM editor "HHED". At this point, there exists a model for all triphones present in the training data. Many of them have only a few examples.

In order to provide sufficient training material for all models, clustering is required [20]-21]. In the HTK, the clustering is accomplished by the HHED tool [22]. In the HTK, the adaptation is performed offline by the HEAdapt using the MLLR technique to estimate a series of transforms or a transformed model set. This reduces the mismatch between the current model set and the adaptation data. The HEAdapt uses a regression class tree to cluster together groups of output distributions that are to undergo the same transformation. The HTK tool HHEd can be used to build a regression class tree and store it as part of the HMM set. A typical use of the HEAdapt involves two passes. On the first pass, a global adaptation is performed. The second pass uses

the global transformation to transform the model set, producing better frame/state alignments which are then used to estimate a set of more specific transforms, using a regression class tree. After estimating the transforms, the HEAdapt can output either the newly adapted model set or the transformations themselves in a transform model file (TMF). The latter can be advantageous if storage is an issue since the TMFs are significantly smaller than the MMFs and the computational overhead incurred when transforming a model set using the TMF is negligible.

IV. Experiment and Results

A. Quranic Phonetic Alphabet (QPA)

The Quranic phonemes are classified into set of classes: Vowels, Glide, Nasal, Voiced and Unvoiced Fricative, Affricative, and Voiced and Unvoiced stop phonemes. Each phoneme has an articulation sign representing the state of the phoneme [1, 9, 15]. They are listed in Table I.

TABLE I
A List of Quranic Monophones, where Monophone(P) and Quranic Phonetic Alphabet (QPA)

	P	QPA		P	QPA		P	QPA
1	ء	ii	12	س	S	23	ل	L
2	ب	b	13	ش	Sh	24	م	M
3	ت	t	14	ص	V	25	ن	N
4	ث	c	15	ض	Vh	26	هـ	H
5	ج	g	16	ط	P	27	و	W
6	ح	x	17	ظ	Ph	28	ي	Y
7	خ	xh	18	ع	J	29	ا	I
8	د	d	19	غ	Jh	30	ـَ	A
9	ذ	dh	20	ف	F	31	ـِ	E
10	ر	r	21	ق	Q	32	ـُ	U
11	ز	z	22	ك	K	33	المد	O

The symbols of the phonemes are designed in this work to be compatible with the Hidden Markov Toolkit (HTK) standards and usage of special characters. These are classified as [15] and [18]:

1. Vowels: [i] {الألف}, [w] {الواو}, and [y] {الياء}. The articulation [o] {المد} can be tied only to the vowels.
2. *Glide Phonemes*: [l] {ل} and [r] {ر}. The [l] {ل} phoneme is called lateral phoneme and [r] {ر} is called trill phoneme.
3. *Nasal Consonants*: [m] {م} and [n] {ن}.
4. *Unvoiced Fricatives Consonants*: [f] {ف}, [c] {ث}, [s] {س}, [sh] {ش}, [v] {ص}, [x] {ح}, [xh] {خ} and [h] {هـ}.
5. *Voiced Fricative Consonants*: [j] {ع}, [jh] {غ}, [z] {ز}, [dh] {ذ} and [ph] {ظ}.
6. *Affricative Consonants*: [g] {ج}.
7. *Voiced Stops Consonants*: [b] {ب}, [d] {د} and [vh] {ض}.
8. *Unvoiced Stops Consonants*: [ii] {ء}, [k] {ك}, [q] {ق}, [t] {ت} and [p] {ط}.
9. *Articulation Signs:* They are [a] {الفتحة}, [e] {الكسرة}, [u] {الضمة} and [o] {المد}.

B. Training and Testing

The process of training the system is divided into four phases: monophones training, fixing the silence model, making triphone models from monophone labels, and triphone clustering. An additional process for adaptation is required. This includes global adaptation phase, regression-class tree phase, and the MLLR adaptation. The training process is repeated twice in every phase for better results.

For the training and testing data, this work used: the 30th Section (called Juza 30) of the Quran, with five of the famous readers of the Quran. They are Ahmad Al-Ajamy, Mohamed Brak, Saa'd Al-Ghamdy, Mashary Al-Ghfary, and Abd-Alrahman Al-Sudees. This Juza includes about 2000 distinct words. It has short verses which are suitable for recognition. Originally, the sound files were recorded using different sampling frequencies-so we sampled them down to 16 kHz. The software, "Sound Forge", was used to segment the sound files into individual files containing the appropriate number of verses-according to the reader's reading style. Based on the segmentation information, the large sound files were divided into 2431 smaller files. The lengths of the verses vary from one to twenty words.

The text was converted into phoneme strings using the Quranic pronunciation rules. A dictionary containing all the words with their transcription is built. The HMM recognizer requires a dictionary containing all the words to be recognized with their transcription. Since there is no such dictionary for Arabic, a new dictionary was built which contains all the words of Juza 30 (for training and testing). A few entries of the dictionary are shown in Table II.

TABLE II
Sample of the Entries in the Structured Dictionary
The (sp) is for Short-Pause

The word in the dictionary	The transcription	The word in Arabic
Jalayhem	*ja la y he m sp*	عليهم
ketaiban	ke ta i ba n sp	كتاباً
Masad	ma sa d sp	مسَد
Beqawle	*be qa w le sp*	بقول
Fawqakum	*fa w qa ku m sp*	فوقكم

The recognition system is based on the HTK version 3.2.1. (Young et al, 2002). The HTK features were used in their standard form. The recognition training and tests were run on a modern standard PC (3.2 GHz processor, 256 MB of RAM) running under Windows XP.

The system is trained for a single reader, Al-Ajamy. Then, the MLLR adaptation algorithm is applied to the remaining readers. Table III shows the results obtained.

TABLE III
THE ACCURACY FOR THE SPEAKER-INDEPENDENT RECOGNIZER. THE FIRST READER "AL-AJAMY" IS CONSIDERED AS THE TRAINING SET. THE OTHER READERS ARE ADAPTED ACCORDING TO THE MLLR

Reader	Accuracy (%)
Ahmad Al-Ajamy	85
Mohamed Brak	78
Saa'd Al-Ghamdy	80
Mashary Al-Ghfary	82
Abd-Alrahman Al-Sudees	68

The results show that a good speaker-independent system can be applied to better fit the characteristics of any new reader. As shown in Table III, the maximum accuracy of the recognizer is 83%, and the lowest is 68%. The reading style of Al-Sudees differs from the other four readers in the reading of the articulation sign [o] {المد}. Al-Sudees is reading this articulation sign with two steps of [o] between words, while the others are using 4 steps. This is the basic reason for the degraded accuracy compared to the other readers.

Table IV shows the run time required for this system. The training process including the four phases took 34 minutes. The adaptation time for every user is about 10 minutes-which is considered a great improvement compared with the training time required for the speaker-dependent systems, which is 34 minutes [19].

TABLE IV
RUN TIME IN MINUTES FOR THE SPEAKER-INDEPENDENT RECOGNIZER RESULTS

Training Process Time in minutes, where each is repeated twice				Adaptation	Recognition Time
Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	
4	4	4	5	5	420

Overall the Speaker-independent system shows a good efficiency and performance in terms of recognition accuracy and total run-time, respectively. A drawback of the system is the recognition time which is needed due to a large number of the states of the HMM model. The size of the states (13 states) is needed to achieve the high accuracy shown in the tables above.

VI. CONCLUSION

This paper introduces a speaker independent Quranic recognizer using the Maximum Likelihood Linear Regression. The MLLR made use of a regression class tree to group Gaussian in a model set. It was used to generate classes dynamically in order to estimate mean transformation and variance transformation matrices.

The process for adaptation includes global, regression class tree (RCT), and MLLR adaptation. Adaptation is performed offline by using the MLLR to estimate a series of transforms. The MLLR makes use of a RCT to group the Gaussians in the model set. This tree is constructed to cluster together components that are close in acoustic space. It is built using the original speaker independent model set, and is thus independent of any new speaker. It is constructed with a centroid splitting algorithm, which uses Euclidean distance measure.

The recognizer system was based on the HTK version 3.2.1. The HEadapt uses the regression class tree for global adaptation and produces a model file. The recognizer uses HMM with thirteen states left-to-right model. The recognizer was trained for a single reader. The MLLR adaptation algorithm was applied to four famous readers of Section 30 in the Quran. The results show that a good speaker independent recognizer can be applied for new readers. The range of accuracy for the tested sample was between 68 and 85 %. These accuracy measures are better than those reached when the speaker dependent tested with the same data set and readers-where the range of accuracy was 49 to 82% [18].

A drawback of the system is the recognition time needed due to a large number of the states of the HMM model. A future research will investigate the system on the other 29 sections in the Quran.

REFERENCES

[1] Al-Diri, B., "A Large Vocabulary Speech Recognition Model for Arabic," *Master Thesis*, University of Jordan, 2002.
[2] Aulama, M., "Arabic Vowel Phonemes Detection and Categorization in Speech Processing," *Master Thesis*, University of Jordan, 2001.
[3] Bahl, L. Balakrishnan, S. Bellegarda, J. Franz, M. Gopalakrishnan, P. Nahamoo, D. Novak, M. Padmanabhan, M. Picheny, M and Roukos, S., "Performance of the IBM Large Vocabulary Continuous Speech Recognition System on the ARPA Wall Street Journal Task," *IEEE Inter. Conf. on Acoustics*, Speech and Signal Processing, vol. 1., pp. 41–44, 1995.
[4] Bateman, D. Bye, D. and Hunt, M., "Spectral Constant Normalization and Other Techniques for Speech Recognition in Noise," *Proc. IEEE. Inter. Conf. Acoustic*. Speech Signal Process, vol.1, pp. 241-244, 1992.
[5] Baum, L.E., "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov processes," *Inequalities*, vol.3, pp. 1-8, 1972.
[6] Christensen, B. Maurer, J. Nash, M. and Vanlandingham, E., "Accessing the Internet via the Human Voice," www.stanford.edu/~jmaurer/homepage.htm, 2001.
[7] Davis S. and Mermelstein P., "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Trans. on Acoustics*, Speech and Signal Processing, vol. ASSP-28, no. 4, pp. 357–366, 1980.
[8] Davis, K. Biddulph, R. and Balashek, S. "Automatic Recognition of Spoken Digits," *Journal of the Acoustical Society of America*, vol.24, pp. 637-642, 1952.
[9] Deller J., Proakis G. and Hansen J., "Discrete-Time Processing of Speech Signals," *The Institute of Electrical and Electronics Engineers Inc.*, New York, 2nd edition, 2000.
[10] Furui S., "Speaker Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum," *IEEE Trans. on Acoustics*, Speech and Signal Processing, vol. 34, no.1, pp. 52–59, 1986.

[11] Gauvain, J. Lamel, L. and Adda-Decker, M. "Developments in Continuous Speech Dictation Using ARPA WSJ Task," *IEEE Inter. Conf. on Acoustics*, Speech, and Signal Processing, vol. 1, pp.65–68, 1995.
[12] Gold B. and Morgan N., "Speech and Audio Signal Processing: Processing and Perception of Speech and Music," *John Wiley & Sons, Inc.*, New York, 2000.
[13] Jelinek, F. "A Fast Sequential Decoding Algorithm Using a Stack," *IBM J. Res. Develop.*, vol.13, pp. 675-685, 1969.
[14] Jelinek, F. Bahl, L. R. and Mercer, R. L., "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," *IEEE Trans. Information Theory*, vol. IT-21, pp. 250-256, 1975.
[15] Majali, S., "A Model for a Limited Domain of Arabic Speech Recognition Using Artificial Neural Network," *Master Thesis*, University of Jordan, 1999.
[16] Markowitz, J., "Using Speech Recognition", *Prentice Hall*, MA, 1st edition, USA, 1996.
[17] Mourtaga, E., M. Abdallah, A. Sharieh, and S. Serahn, "Quranic Based Speaker-Dependent Recognition Using Triphone/HMM Model," accepted in AMSE, 2005.
[18] Pallett, D. Fiscus, J. Fisher, W. Garofolo, J. Lund, B. Martin, A. and Przybocki, M., "1994 Benchmark Tests for the ARPA Spoken Language Program," *DARPA Spoken Language Systems Technology Workshop*, pp. 5–36, 1995.
[19] Rabiner L., "Fundamentals of Speech Recognition," *PTR Prentice-Hall Inc.*, New Jersey, 1993.
[20] Ursin, M., "Triphone Clustering in Finish Continuous Speech Recognition," *Master Thesis*, Helsinki University of Technology, 2002.
[21] Woodland, P. Leggetter, C. Odell, J. Valtchev, V. and Young, S., "The 1994 HTK Large Vocabulary Speech Recognition System," *IEEE Inter. Conf. on Acoustics*, Speech, and Signal Processing, vol.1, pp.73–76, 1995.
[22] Young, S. Evermann, G. Hain, T. Kershaw, D. Moore, G. Odell, J. Ollason, D. Povey, D. Valtchev, V. Woodland, P., "The HTK Book (for HTK Version 3.2.1)," *Cambridge University*, Engineering Department, 2002.

Ahmad Sharieh was born in Amman/Jordan 10th of Dec. 1954. Sharieh received his BS degree in Mathematics from The University of Jordan Amman/Jordan 1977, BS in Computer Science from The University of Tennessee Martin/TN USA 1983, MSc in Computer Science from Western Kentucky University Bowling Green/KY USA 1985, and PhD in Computer and Information Sciences from Florida State University Tallahassee/FL USA 1991. He worked as high school Math. teacher (1977-1981), teaching and research assistant at The University of Jordan (1985-1987), assistant professor at Fort Valley College GA/USA (1991-1992), assistant prof. and associate prof. at The University of Jordan (1992-2004), and DEAN of King Abdullah School for Information Technology at The University of Jordan (2004-present).

He authored and participated in authoring of six books, in Arabic and English, including: Computer and Software Packages 6th Ed., Amman, Jordan, Dar WAEL 2005, Learn C++, Amman, Jordan, Dar Almaseera, 2005, and Computer Skills2, Amman, Jordan, Dar WAEL 2006. He published more than 30 papers in journals and conferences. His research interest is in Operating Systems, Parallel Processing, Pattern Recognition, Simulation, and Software Engineering.

Dr. Sharieh provides services to the Ministry of Education, and Ministry of Higher Education in Jordan as IT consulting. He was awarded Hesham Adeeb Hejawee Prize for Applications Sciences: Communication and Information Technology Sector, The 8th Period, Dec. 17th, 2000, Amman, Jordan and awarded the prize for the best authored book from The University of Jordan, for his book Computer and Software Packages 2001.