

# Automatic Speech Annotation Using HMM based on Best Tree Encoding (BTE) Feature

Amr M. Gody<sup>\*1</sup>, Rania Ahmed Abul Seoud<sup>\*2</sup>, Mohamed Hassan<sup>\*3</sup>

*\*Electrical Engineering, Faculty of Engineering, Fayoum University  
Egypt*

<sup>1</sup> amg00@fayoum.edu.eg

<sup>2</sup> r-abulseoud@k-space.org

<sup>3</sup> mh1323@fayoum.edu.eg

**Abstract:** *Manual annotation for time-aligning a speech waveform against the corresponding phonetic sequence is a tedious and time consuming task. This paper aimed to introduce a completely automated phone recognition system based on Best Tree Encoding (BTE) 4-point speech feature. BTE is used to find phoneme boundaries along speech utterance. Comparison to Mel-frequency cepstral coefficients (MFCCs) speech feature in solving the same problem is provided. Hidden Markov Model (HMM) and Gaussian Mixtures are used for building the statistical models through this research. HTK software toolkit is utilized for implementation of the model. The System can identify spoken phone at 59.1% recognition rate based on MFCC and 22.92% recognition rate based on BTE. The current BTE vector is 4 components compared to 39 components of MFCC. This makes it very promising features vector, BTE with 4 components gives a comparable recognition success rate compared to the 39 components MFCC vector widely in the area of ASR.*

**Key words:** *BTE, MFCC, HTK, Gaussian Mixture, Speech Recognition.*

## 1 INTRODUCTION

Presently, manual annotation by expert phoneticians is the most precise way for time-aligning a speech waveform against the corresponding phonetic sequence. This is a tedious and time consuming task, which makes it a prohibitive choice for large speech corpora. Several approaches have been proposed for the task of speech segmentation [2-6]. The most frequently used approach is based on HMM phone models. In this method each speech waveform is initially decomposed into a sequence of feature vectors, using a speech parameterization technique. Afterwards, a set of HMM phone models (phone recognizer) is utilized to extract the corresponding phonetic sequence as well as the positions of the phonetic boundaries. Other speech segmentation methods have also been proposed in the literature. Some of them include detection of variations/similarities in spectral or prosodic parameters of speech, template matching using dynamic programming and/or synthetic speech and discriminative learning segmentation.

Various speech parameterizations have been utilized in the phonetic segmentation task, with the Mel Frequency Cepstral Coefficients (MFCC) among the most widely used, especially in the HMM-based approach. Other speech features such as Perceptual Linear Prediction (PLP), Line Spectral Frequencies (LSF), Linear Predictive Coding (LPC), short-time energy, formants and wavelet-based have also been used.

Automatic annotation is used to make a preliminary solution before starting the manual annotation. Its task is to simplify the effort in the manual annotation task. In this paper, the most frequently approach – adapting a Hidden Markov Model (HMM) based phonetic recognizer to the task of automatic phonetic segmentation is used. Our base line system contains 10ms frame rate with 25ms Hamming window. Here the speech is parameterized using MFCC and BTE. MFCC with 12 Mel-Frequency Cepstral Coefficients and normalized log energy, as well as their first and second order differences yielding a total of 39 components. Another parameterization technique is Best Tree Encoding BTE with 4 spectral based components. A set of context-independent Left -To -Right (LR) monophone HMMs with one Gaussian per state are flat-initialized. The HMM model is 3 emitting states. These HMMs are well trained using HMM Tool Kit (HTK) and both features MFCC and BTE for the problem of automatic annotation.

Speech database is prepared to measure the quality of this experiment. Speech database is labeled and transcribed then verified to evaluate the results of automatic segmentation. The following sections will navigate through the details of this research. Section 2 will illustrate problem definition. In section 2, the HMM GMM based speech recognition will be illustrated. BTE speech feature is explored in section 3. The experimental Framework will be provided in section 4. The experimental procedure will be presented in section 5. The results will be presented in section 6. The conclusion will be given in section 7. Then finally the list of references will be listed in section 8.

## 2 PROBLEM DEFINITION

Automatic Speech annotation to Arabic phone level is the problem that is intended in this research. The phone is supposed to be the basic speech unit. Finding the phone boundaries along the stream of human speech is the basic definition of the annotation. Speech features should be stable along the phone duration. The best the features are the accurate the boundaries are.

## 3 HMM–GMM BASED SPEECH RECOGNITION

In HMM–GMM (Hidden Markov Model –Gaussian Mixture model related) based speech recognition ,see [Gales and Young, 2007](#) for review[10], the short-time spectral Characteristics of speech is turned into a vector (the “observations” of Fig. 1, sometimes called frames), and build a generative model using a HMM that produces sequences of these vectors. A left-to-right three-state HMM topology as in Fig. 1 will typically model the sequence of frames generated by a single phone. Models for sentences are constructed by concatenating HMMs for sequences of phones. Different HMMs are used for phones in different left and right phonetic contexts, using a tree-based clustering approach to model unseen contexts, see [Young et al., 1994](#) for review [11]. The index  $j$  will be used for the individual context-dependent phonetic states, with  $1 \leq j \leq J$ . While  $j$  could potentially equal three times the cube of the number of phones (assuming only the immediate left and right phonetic context will be modeled), after tree-based clustering it will typically be several thousand. The distribution that generates a vector within HMM state  $j$  is a Gaussian Mixture Model (GMM):

$$P(x|j) = \sum_{i=1}^{M_j} w_{ji} N(x, \mu_{ji}, \Sigma_{ji}) \quad (1)$$

Table 1 shows the parameters of the probability density functions (pdfs) in an example system of this kind: each context dependent state (of which we only show three rather than several thousands) has a different number of sub-states  $M_j$ .

TABLE 1: PARAMETERS FOR PDFS IN GMM HMM SYSTEM

State 1	State 2	State 3
$\mu_{11}, \Sigma_{11}, w_{11}$	$\mu_{21}, \Sigma_{21}, w_{21}$	$\mu_{31}, \Sigma_{31}, w_{31}$
$\mu_{12}, \Sigma_{12}, w_{12}$	$\mu_{22}, \Sigma_{22}, w_{22}$	$\mu_{32}, \Sigma_{32}, w_{32}$
$\mu_{13}, \Sigma_{13}, w_{13}$	$\mu_{23}, \Sigma_{23}, w_{23}$	
	$\mu_{24}, \Sigma_{24}, w_{24}$	

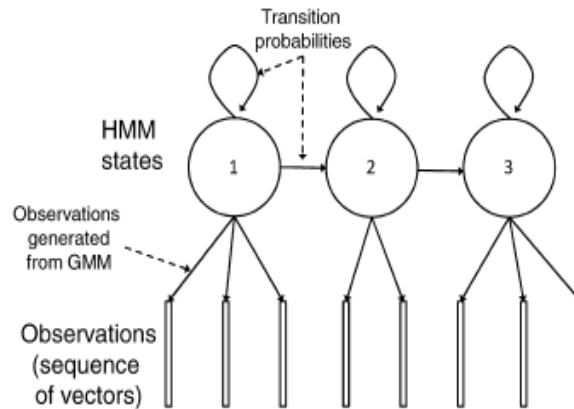


Figure1: HMM for speech recognition

HTK is principally concerned with continuous density models in which each observation probability distribution is represented by a mixture Gaussian density. In this case, for state  $j$  the probability  $b_j(o_t)$  of generating observation  $o_t$  is given by

$$b_j(o_t) = \prod_{s=1}^S \left[ \sum_{m=1}^{M_{js}} c_{j_{sm}} N(o_{st}; \mu_{sm}, \Sigma_{j_{sm}}) \right]^{y_s}$$

where  $M_{js}$  is the number of mixture components in state  $j$  for stream  $s$ ,  $c_{j_{sm}}$  is the weight of the  $m$ 'th component and  $N(o; \mu, \Sigma)$  is a multivariate Gaussian with mean vector  $\mu$  and covariance matrix  $\Sigma$ , that is

$$N(\mathbf{o}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{o}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{o}-\boldsymbol{\mu})}$$

where  $n$  is the dimensionality of  $\mathbf{o}$ . The exponent is a stream weight and its default value is one.

Other values can be used to emphasize particular streams, however, none of the standard HTK tools manipulate it. HTK also supports discrete probability distributions in which case

$$b_j(\mathbf{o}_t) = \prod_{s=1}^S \{P_{js}[v_s(\mathbf{o}_{st})]\}$$

where  $v_s(\mathbf{o}_{st})$  is the output of the vector quantiser for stream  $s$  given input vector  $\mathbf{o}_{st}$  and  $P_{js}[v]$  is the probability of state  $j$  generating symbol  $v$  in stream  $s$ . In addition to the above, any model or state can have an associated vector of duration parameters  $\{dk\}$ . Also, it is necessary to specify the kind of the observation vectors, and the width of the observation vector in each stream. Thus, the total information needed to define a single HMM is as follows

- Type of observation vector
- Number and width of each data stream
- Optional model duration parameter vector
- Number of states
- For each emitting state and each stream
  - Mixture component weights or discrete probabilities
  - If continuous density, then means and covariance
  - Optional stream weight vector
  - Optional duration parameter vector
- Transition matrix

In automatic speech recognition (ASR) systems, it is normally used Gaussian mixture HMMs as acoustic models for modeling basic speech units, ranging from context-independent whole words in small vocabulary ASR tasks to context-dependent phonemes (e.g., triphones) in large vocabulary ASR. Traditionally, the HMM-based acoustic models are estimated from available training data using the well-known EM algorithm based on the maximum-likelihood (ML) criterion. To deal with data sparseness problems in model training, we normally use phonetic decision trees to tie HMM states from different triphone contexts. In order to derive a simple closed-form solution, we normally grow the decision trees based on simple models, such as single Gaussian HMMs. After the state-tied structure is determined from the decision trees, a separate “mixing-up” step is used to gradually increase the number of Gaussian mixtures in each tied HMM state until the optimal performance is achieved. In today’s ASR systems, e.g., HTK, “mixing-up” is normally implemented in two steps [2]:

- 1) All existing Gaussians or the most dominant Gaussian mixture component in an HMM state is split based on some random or heuristic strategies.
- 2) All split Gaussians are re-estimated based on the EM algorithm.

Obviously, this incremental method for increasing model complexity is a good strategy to learn very large-scale statistical models without getting trapped in any bad local optimum. However, we still face some problems when increasing model complexity in the above “mixing-up” strategy. First of all, the random splitting strategy is not optimal in terms of the model estimation criterion. For example, there is no guarantee that the newly added Gaussian components from random splitting always increase the likelihood function prior to re-estimation. Second, since the subsequent EM-based re-estimation is sensitive to the initial parameters of the randomly split Gaussians, there is no guarantee that the EM-based re-estimation can always converge to the optimal point.

In HTK, the conversion from single Gaussian HMMs to multiple mixture component HMMs is usually one of the final steps in building a system. The mechanism provided to do this is the HHED MU command which will increase the number of components in a mixture by a process called *mixture splitting*. This approach to building a multiple mixture component system is extremely flexible since it allows the number of mixture components to be repeatedly increased until the desired level of performance is achieved.

The MU command has the form

MU n itemList

where  $n$  gives the new number of mixture components required and `itemList` defines the actual mixture distributions to modify. This command works by repeatedly splitting the mixture with the largest mixture weight until the required number of components is obtained. The actual split is performed by copying the mixture, dividing the weights of both copies by 2, and finally perturbing the means by plus or minus 0.2 standard deviations. For example, the command has the form

```
MU n itemList
```

For example, the command

```
MU 3 {aa.state[2].mix}
```

would increase the number of mixture components in the output distribution for state 2 of model `aa` to 3. Normally, however, the number of components in all mixture distributions will be increased at the same time. Hence, a command of the form is more usual

```
MU 3 {*.state[2-4].mix}
```


It is usually a good idea to increment mixture components in stages, for example, by incrementing by 1 or 2 then re-estimating, then incrementing by 1 or 2 again and re-estimating, and so on until the required number of components is obtained. This also allows recognition performance to be monitored to find the optimum.

We can start prototype of phone in HMM with 4 mixtures per state. However, this was (a pretty good) guess of us. To be sure that we have chosen the optimal topology for our models there is no way to avoid the heuristic try-and-fail method. We ran a series of trainings on different number of mixtures. It is recommended to start with a single Gaussian model, train it until it converges on the dev set and then increase the number of mixtures by one, train them and so on.

One final point with regard to multiple mixture component distributions is that all HTK tools ignore mixture components whose weights fall below a threshold value called `MINMIX` (defined in `HModel.h`). Such mixture components are called *defunct*. Defunct mixture components can be prevented by setting the `-w` option in `HEREST` so that all mixture weights are floored to some level above `MINMIX`. If mixture weights are allowed to fall below `MINMIX` then the corresponding Gaussian parameters will not be written out when the model containing that component is saved. It is possible to recover from this, however, since the `MU` command will replace defunct mixtures before performing any requested mixture component increment.

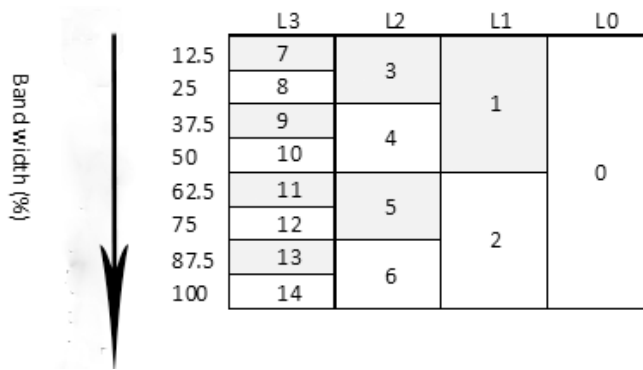
#### 4 BEST TREE ENCODING

BTE is a simple on/off entropy mapping of the signal into the bands in which the signal is decomposed using wavelet packets. The key property in BTE is the alignment of the neighboring frequency domain bands in wavelet packets decomposition of the signal. Adjacent bands are much closer in distance than the non adjacent bands.



	L3	L2	L1	L0
12.5	0	2	6	14
25	1			
37.5	3			
50	4	5		
62.5	7	9	13	
75	8			
87.5	10	12		
100	11			

Part a: Before BTE



Part b: After BTE

Figure 2: BTE bands are aligned such as to make adjacent wavelet bands are closer in distance than non adjacent bands.

Figure 2-a illustrates how bands are sorted according to Matlab wavelet packets function. Figure 2-b indicates how bands are encoded in BTE. Bands are rearranged for calculating the BTE of the frame. The tree is Encoded into a single number that held information of tree structure {leaves} and weight according to figure 2-b.

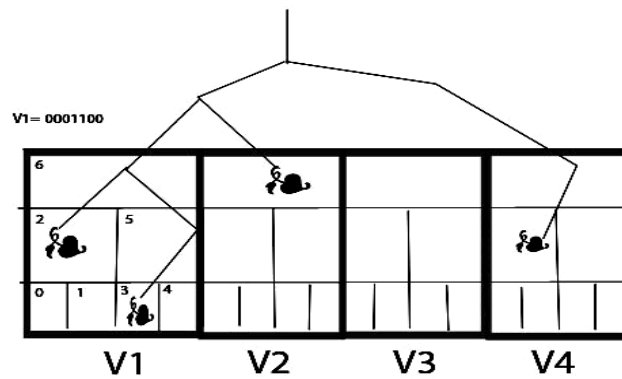


Figure 3: BTE for certain wavelet packets Best tree structure

The indicated tree structure in figure 3 will be encoded into features vector of 3 elements as shown in table 2.

TABLE 2: BEST TREE 4 POINT ENCODING EVALUATION

Element	Binary Value	Decimal value	Frequency Band
V1	0001100	12	0 - 25 %
V2	1000000	64	25% - 50%
V3	0000000	0	50%-75%
V4	0000100	4	75%- 100%

Features BTE vector  $\zeta$  for this example of speech frame will be  $\zeta = \begin{bmatrix} 12 \\ 64 \\ 0 \\ 4 \end{bmatrix}$

## 5 EXPERIMENT FRAMEWORK

The framework we developed to train and test GMM HMM models uses HTK to do feature extraction and build the baseline models which are used to align the training data. Microsoft C# (C sharp) is used for building the needed programs and algorithms for building initial models of HTK. HTK tools for training and decoding is a collection of command-line options such as HERest and HVite. Each makes a special function, which is explained in detail in HTK book [9]

The phonetic context tree of the HTK baseline models is utilized in the proposed system. Training and testing in the proposed system is based on Weighted Finite State. HTK tools evaluate the Viterbi path based on likelihood.

## 6 AUTOMATIC ANNOTATION EXPERIMENTAL PROCEDURE

### A. Database Preparation

- a. Corpus of 300 Arabic sentences of 30 persons (males) sampling rate of 32 kb/s is used. All samples are manually annotated.
- b. The Database is split into two groups of 150 sentences each. Group A is for training and Group B is for testing.

### B. Features Extraction

- a. All samples are processed to generate MFCC -39 points feature. HTK is used in this step.
- b. All samples are processed to generate BTE -4 points feature. Matlab is used in this step.

### C. Marshaling

All feature files are normalized for being processed in HTK. This process is called marshaling. The data from different sources are rearranged in a way that to be understood by HTK tools. BTE feature vectors files are marshaled into HTK format. HTK allows for user defined features type. This will give HTK tools the ability to be used to process data from other sources not just HTK tools.

### D. Model Design

- a. Five nodes LR HMM model is created to model a single phone.
- b. Survey for the most frequently used Gaussian Mixture count for MFCC is used to set the number of Gaussian Mixtures of MFCC model.
- c. For BTE; Gaussian mixture count is an experiment parameter. It will be tuned for the best success rate.
- d. Dictionary and Grammar files will be created for HTK phone recognition problem.  
{Illustrate the Grammar file and the dictionary by a graph and a table that clarify the Grammar network and the dictionary}

### E. Training the Models.

- a. Using HTK and the training samples for MFCC, MFCC models will be trained.
- b. Using HTK and the training samples for BTE, BTE models will be trained.

### F. Testing the Models.

- a. Using HTK and the testing samples for MFCC, MFCC models will be tested.
- b. Using HTK and the testing samples for BTE, BTE models will be tested.

### G. Results

- a. Results are tabulated for MFCC based recognizer.
- b. Results are tabulated for BTE based recognizer.

Table 3 illustrates the results obtained from both systems. As of the results BTE-4 indicates very comparable results to the well known MFCC features. BTE is still in the development phase. This makes it very promising. BTE is 4 components compared to 39 components of MFCC, makes it a very promising feature.

TABLE 3: BTE-4 VERSES MFCC-39 RECOGNITION RESULTS

Feature Type	% Correct	N	I	D	S
BTE-4	22.92%	22542	61373	307	17069
MFCC-39	59.1%	19455	71637	204	7754

N: the total number of labels in the reference transcriptions

I: Number of Insertions errors in the results string.

D: Number of deletion errors in results string.

S: Number of substitution errors in results string.

The number of GM is a factor in the success rate for BTE experiment. This number is altered as an experiment parameter. Figure 4 gives the results of changing this value on the success rate.

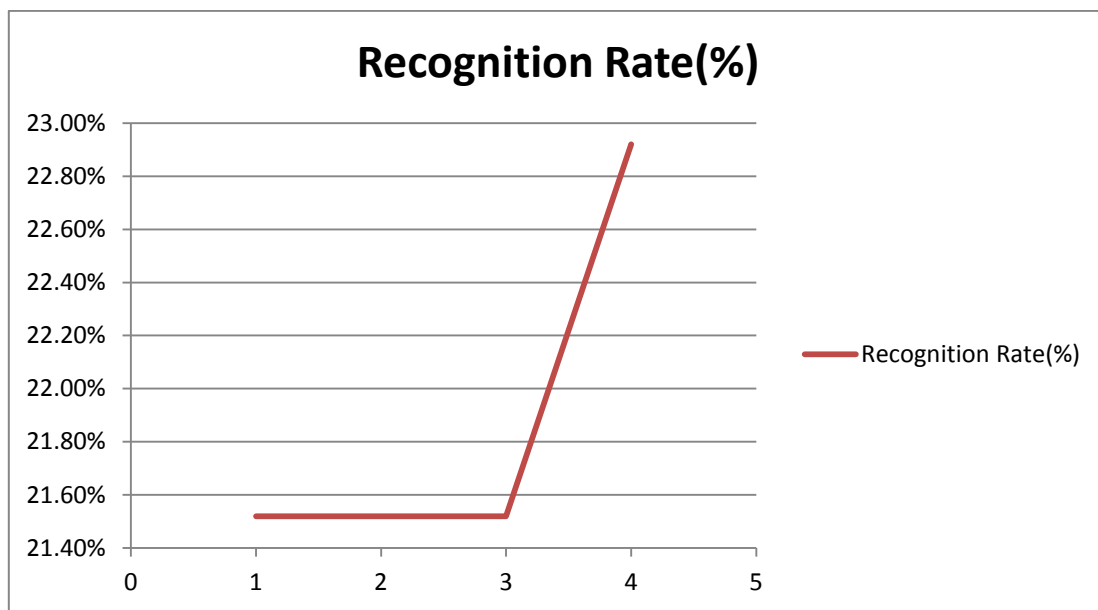


Figure 4: Recognition Rate versus Max Number of Mixtures

## 7 CONCLUSIONS

The results tabulated in table 1 indicate that BTE with 4 components is very promising. BTE is newly developed feature that relies on the spectral information. It is composed of 4 components that are used to encode the whole spectral information of the signal. It gives very close results to the well known feature MFCC with 39 components. This makes it a very promising enhancement that gives much more efficient results than MFCC.

## REFERENCES

- [1] Amr M. Gody, "Wavelet Packets Best Tree 4-Points Encoded (BTE) Features", the 8th Conference on Language Engineering, 2008, PP. 189-198, Cairo, Egypt.
- [2] Iosif Mporas, Todor Ganchev, Nikos Fakotakis, "Phonetic segmentation using multiple speech features", International Journal of Speech Technology, Springer Netherlands, Volume 11, Number 2 / June 2008, PP. 73-85
- [3] Kris Demuynck, Tom Laureys, "A Comparison of Different Approaches to Automatic Speech Segmentation", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 2448/2006, ISBN 978-3-540-44129-8, PP. 385-406
- [4] Z. M. šarić, S. R. Turajlić, "A new approach to speech segmentation based on the maximum likelihood", Journal of Circuits, Systems, and Signal Processing, Birkhäuser Boston, Volume 14, Number 5 / September 1995, PP. 615-632
- [5] Chin-Teng, Der-Jenq, Rui-Cheng, Gin-Der, "Noisy Speech Segmentation/Enhancement with Multiband Analysis and Neural Fuzzy Networks", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 2275/2002, ISBN 978-3-540-43150-3, PP. 81-94.

[6] Yanxiang Chen, Qiong Wang, "A Speaker Based Unsupervised Speech Segmentation Algorithm Used in Conversational Speech", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 4798/2007, ISBN 978-3-540-76718-3, PP. 396-402.

[8] Amr M. Gody, "Voiced/Unvoiced and Silent Classification Using HMM Classifier based on Wavelet Packets BTE features", the 8th Conference on Language Engineering. 2008, Cairo, Egypt.

[9] Steve Young, Mark Gales, Xunying Andrew Liu, Phil Woodland, et al., 2006. The HTK Book, Version 3.41, Cambridge University Engineering Department, <http://www.htk.eng.cam.ac.uk>.

[10] Gales, M.J.F., Young, S.J., 2007. *The application of hidden Markov models in speech recognition*. Foundations and Trends in Signal Processing (3), 195–304.

[11] Young, S., Odell, J.J., Woodland, P.C., 1994. *Tree-based state tying for high accuracy acoustic modeling*. In: Proc. 1994 ARPA Human Language Technology Workshop, pp. 304–312.

## BIOGRAPHY



**Amr M. Gody** received the B.Sc. M.Sc., and Ph.D. from the Faculty of Engineering, Cairo University, Egypt, in 1991, 1995 and 1999 respectively. He joined the teaching staff of the Electrical Engineering Department, Faculty of Engineering, Fayoum University, Egypt in 1994. He is author and co-author of about 40 papers in national and international conference proceedings and journals. He is the Acting chief of Electrical Engineering department, Fayoum University in 2010, 2012 till now. His current research areas of interest include speech processing, speech recognition and speech compression.



**Rania Ahmed Abul Seoud** received the B.S. degrees in Electrical Engineering- Communications and Electronics Department at Cairo University – EL Fayoum Branch in 1998 and M.S.E. degrees in Computer Engineering at Cairo University in 2005. Her Ph.D. degree was from the Biomedical Engineering department, Cairo University in 2008. She worked as a Demonstrator and a Teaching Assistant in Electrical Engineering Department of Misr University for Science and Technology, Egypt since 1998. Currently, she is a lecturer in Electrical Engineering Department of EL Fayoum University, Egypt. Her areas of interest in research are Artificial Intelligence, Natural Language Processing, computational linguistics, machine translation, application of artificial Intelligence to computational biology and bioinformatics and Computer networks.

## التمييز التلقائي للكلام باستخدام نموذج ماركوف الخفي المعتمد على الخواص المستنتجة من تشفير الشجرة المثلى (BTE)

عمرو محمد جودي، رانيا احمد ابو السعود، محمد حسن  
قسم الهندسة الكهربائية، كلية الهندسة، جامعة الفيوم

### خلاصة:

التحديد اليدوي للصوتيات المنطوقة (الفون) تعتبر عملية صعبة و مكلفة جدا من ناحية الوقت. هذا البحث يقدم محاولة لحل هذه المشكلة باستخدام الخواص المستنتجة من تشفير الشجرة المثلى (BTE-4). هذه الخواص استخدمت لتحديد الحدود الزمنية لوحدات الصوت المنطوقة من متحدث. مقارنة مع واحدة من الخواص الشهيرة المستخدمة في هذا المجال (MFCC) قد قدمت للقارئ. نموذج ماركوف الخفي (HMM) المدعوم بنموذج احصائي معروف بخليط جاوس قد استخدم لتوصيف الوحدات الصوتية. و قد استخدم برنامج معروف لبناء نماذج ماركوف لوحدات الصوت و هو (HTK). و بمعمل المقارنة قد كانت نتائج التعرف الصحيح باستخدام MFCC ٥٩,١% بينما كانت النتائج باستخدام BTE ٢٢,٩٢%. و لكن بمقارنة عدد العوامل في متجه الخواص المعتمد على BTE فهو ٤ عوامل بينما هذا العدد هو ٣٩ بالنسبة لمتجه الخواص المعتمد على MFCC. نتائج متجه BTE الى نتائج متجه MFCC هي ٣٩% بينما حجم متجه BTE الى حجم متجه MFCC هو ١٠% و هذا الفرق في حجم متجه الخواص بالنسبة للفرق في النتائج يجعل متجه BTE واعد جدا للاستخدام في تطبيقات التعرف الالي على الاصوات حيث يمكن اضافة عوامل اخرى لمتجه الخواص المعتمد على BTE للحصول على نتائج افضل مع الاحتفاظ بالتنافسية الحجمية مع متجه خواص MFCC .