International Conference on Engineering Research & Applications (ICERA)
Istanbul (Turkey) May. 17-18, 2017

Investigation of HTK for Arabic Phonemes Boundary Detection

Abed Ahcène, Amrouche Aissa and Boubakeur Khadidja Nesrine

Scientific and Technical Research Center for the Development of the Arabic Language, Algeria
abedahcene@gmail.com ; amrouche_a@yahoo.fr ; boubakeur.khadidja@gmail.com

***Abstract*:** *In this paper we propose an automatic Arabic phonemes boundary detection system. This system is mainly used to perform an automatic speech corpus labeling. Because the manual labeling is a hard task and consumes time. We have used the HTK (Hidden Markov Tools Kit) model to solve this problem. The Hidden Markov Models implementation is used to detect phonemes boundaries with the textual information given by the transcription file. The final system improves a Correct Classification Rate of 89.5%, obtained by 5-HMM of 8 Gaussian components.*

Keywords: *HTK, HMM, Arabic language, Phonemes Boundary Detection, Speech corpus labelling.*

1. Introduction

The speech corpus design is organized in three main stages: textual material construction which is presented as words and phrases ; corpus recording by spanning peoples with multiple range ages and regions, and the labelling corpus. The speech signals must be segmented into phonemes. The aim of this paper is the use of Hidden Markov Tools Kit (HTK) to detect the Arabic phonemes boundaries.

Speech segmentation is the process to detect phonemes boundaries of a speech signal. Several methods are investigated in this field [1]. The most speech segmentation methods are based to force the alignment which requires the phonetic transcription materials [2]. These methods use the dynamic programming and Hidden Markov Models (HMM) to estimate the best path in features parameters space [3], [4]. In addition, an automatic Arabic speech segmentation system is proposed by detecting the Delta-MFCC peaks of speech signal [5].

Initially, we proposed an Automatic speech Segmentation into Phonemes (ASPh) for the Arabic language. This system is implemented using the HMM under Matlab environment, which shows a Correct Classification Rate (CCR). To improve the rate an Artificial Neural Networks (ANN) is used with a CCR [6]. However, its outputs give the frame class, and our focus is to estimate the boundary of each phoneme.

To solve these problems, our researches are oriented towards the ASPh implementation under the HTK environment [7]. It is a tool for building and manipulating hidden Markov models. HTK is mainly used for automatic speech recognition, although it has been used for many other applications, including speech synthesis ; character recognition and several researchers in the worldwide.

This paper is organized as follow : in the first part we have studied the place and mode of Arabic phonemes articulation. The second part is reserved to the system implementation introducing the database creation and the recognition phase. Finally, we have studied the system evaluation performances versus Gaussian components number and hidden states number.

2. Modern Standard Arabic Background

The Modern Standard Arabic (MSA) consists of 40 phonemes: 28 consonants, 6 vowels (3 long and 3 short vowels) and 6 vocal variants in emphatic context. Each phoneme differs from the other by the place and articulation mode. Generally, there are 16 places of articulation (Figure 1).

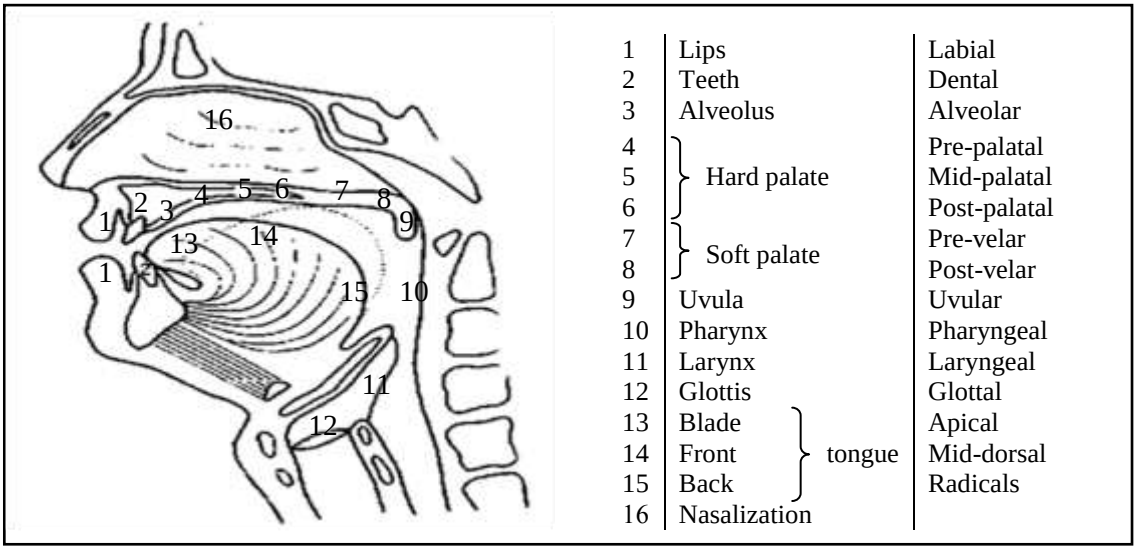

Fig. 1: Articulation places of Arabic phonemes [8]

The MSA is characterized of two phonemes classes. The first class includes four emphatic consonants, [sˤ], [dˤ], [tˤ] and [ðˤ] which are the consonants emphatic versions of [s], [d], [t] and [ð]. The second class contains five pharyngeals : two fricatives [ħ], [ʔ] and three uvulars [χ], [ʁ] and [q] [9]. The Table 1 shows the places and modes of arbic phonemes articulation.

TABLE I: Arabic phonemes articulation [10]
V : Voiced ; U : Unvoiced

	Plosives		Nasals		Fricatives		Liquids		Semivowels	
	V	U	V	U	V	U	V	U	V	U
Bilabials	ب [b]		م [m]						و [w]	
Labio-dentals						ف [f]				
Inter-dentals					ذ [ð]	ث [θ]				
					ظ [ðˤ]					
Alveolars	د [d]	ت [t]	ن [n]		ز [z]	س [s]	ر [r]			
		ط [tˤ]			ض [dˤ]	ص [sˤ]	ل [l]			
Alveo-palatals						ش [ʃ]			ي [j]	
Palatals					ج [ʤ]					
Velars		ك [k]								
Uvulars		ق [q]			غ [ʁ]	خ [χ]				
Pharyngeals					ع [ʕ]	ح [ħ]				
Glottals		ء [ʔ]			ه [h]					

3. Methods and Materials

The proposed system is presented as a graphical interface implemented under the Matlab environment. This application is decomposed into two main phases : the reference phase creation of and the segmentation phase. The figure 2 shows the ASPh system interface.

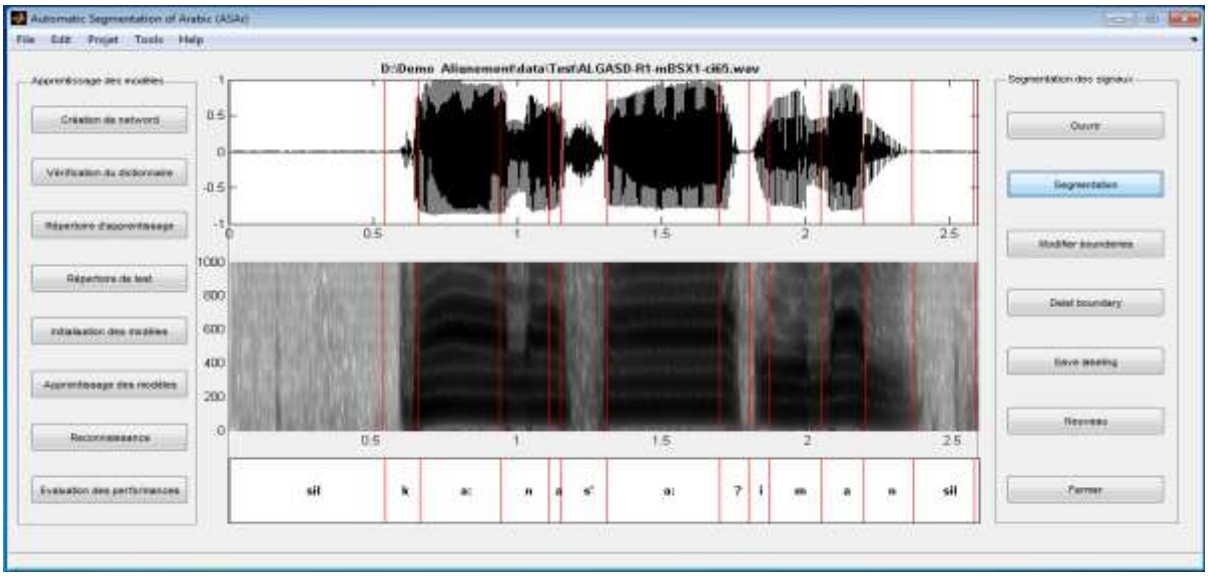

Fig. 2: ASPh System Interface

Several steps are implemented to build the ASPh system. These steps are summarized as follows :

- Data preparation ;
- Features extraction ;
- Models prototype and initialization ;
- System learning ;
- Recognition phase;
- Performances evaluation.

3.1. Data preparation

The first step is the grammar file construction, to define the constraints of the system target input. The grammar file is created with :

> *$ph = ? | b | t | T | Z | X\ | X | d | D | r | z | s | S | s' | d' | t' | D' | ?' | G | f | q | k | l | m | n | h | w | j | a | a: | i | i: | u | u: ;*
> *({START_SIL } { $ph } {END_SIL})*

The grammar file conversion into the HTK word trellis network is obtained by using the '*HParse'* tool :

> *$ HParse grammaire wordnet*

The dictionary provides the match between the phonemes used in the grammar file and the acoustic models. In this task the phonemic models is used to simplify the dictionary structure which is represented as follows :

> *? [?] ?*
> *b [b] b*
> *t [t] t*
> .
> *START_SIL [sil] sil*
> *END_SIL [sil] sil*

3.2. Features extraction

This is the step of extracting acoustic parameters MFCC (Mel Frequency Cepstral Coefficients). This parameterization must be compatible with the HTK tool, the nature of the audio data (format, sampling

frequency, etc.) and parameters characteristics (parameter type, window length, pre-emphasis, etc.). To do this, we created a configuration file as follows:

```
# Coding parameters
SOURCEKIND     = WAVEFORM
SOURCEFORMAT   = WAV
SOURCERATE     = 625
TARGETKIND     = MFCC_E_D_A
TARGETRATE     = 100000
WINDOWSIZE     = 250000
USEHAMMING     = TRUE
PREEMCOEF      = 0.97
NUMCHANS       = 26
NUMCEPS        = 12
ENORMALISE     = TRUE
```

It is possible to modify certain parameters according to our needs. Then, an HTK script file '*Hcopy.scp'* is created which contains the following lines :

```
../train/S1.WAV ../train/S1.mfc
../train/S2.WAV ../train/S2.mfc
..
```

Each file in the learning corpus is represented by a row. The *'Hcopy.scp'* script tells the HTK to extract the acoustic parameters for each audio file of the first column, then save the results in the second column of corresponding file. The responsible command is :

```
$ HCopy -T 1 -C config -S hcopy.scp
```

3.3. Models prototype and initialization

An HMM prototype (*'proto'*) is created, left-right with 3 states and 39 acoustic parameters vectors, as follows:

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
<NumStates> 5
<State> 2
<Mean> 39
0.0 0.0 0.0 ...
<Variance> 39
1.0 1.0 1.0 ...
.
<TransP> 5
0.0 1.0 0.0 0.0 0.0
0.0 0.6 0.4 0.0 0.0
0.0 0.0 0.6 0.4 0.0
0.0 0.0 0.0 0.7 0.3
0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

The HTK tool offered the '*Hinit'* module, to initialize the prototype model with the means and variances of the learning data. Each state of the HMM has several Gaussian components, the learning vectors are associated with the most likely Gaussian component. The vectors number associated with each component in a state can be used to estimate the weights of the mixtures. The *'Hinit'* structure command is given by :

```
$ HInit hmm data1 data2 data3
```

3.4. System Learning

The final model of the HMM is estimated by applying the *'HRest'* command to the generated model by *'Hinit'*

```
HERest -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm0\macros -H hmm0\hmmdefs -M hmm1 mphone0
HERest -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp -H hmm1\macros -H hmm1\hmmdefs -M hmm2 mphone0
.
.
```

This command creates an HMM for each phoneme given by the file *'phones0.mlf'* from the template stored in the folder *'hmm0'*. The resulting models must be stored in the folder *'hmm1'*. This procedure will be repeated three or four times.

3.5. Recognition Phase

A recognition phase is required after model creation,. This is done by executing the command *'Hvite'*, when *'testf.mfc'* is the speech signal.

```
HVite -H macros -H hmmdefs -i phnfile.phn -I word.mlf -S testf.mfc dict mphone1
```

4. Results and Discussion

The ALGASD database is used to calculate the performance of ASPh. 288 signal belong to a region are taken from the global corpus. 202 (70%) of them are used for the learning corpus, and the remainder of signals are used in the test phase.

TABLE II : ASPh performances vs. Gaussian components numbers

	Number of Gaussian Components			
	2	4	8	16
3-HMM	81.5%	84.75%	88.35%	88.37%
5-HMM	83.42%	86.56%	89.5%	89.45%

Table II shows the correct classification rates obtained for the Gaussian Components numbers function (CG). The best correct classification rate is 89.45% obtained by the system using 5-HMM and 8 Gaussian components. In general, with 5-HMM, the system is more efficient than 3-HMM. This means that a remarkable improvement when the Artificial Neural Networks are implemented [1].

5. Conclusion

In this work, we have presented the different steps to implement the automatic detection system for Arabic phonemes boundaries. Two main phases are carried out: the learning phase and the recognition phase. The learning phase is implemented using different modules of the HTK tool. The obtained results demonstrate the powerful of using the HTK to solve the speech segmentation problem. However, even with this system the final segmentation must be verified by the user.

The implementation of the ASPh is almost complete. We seek to improve the performance by increasing the amount of learning data. The use of other methods of features extraction such as Linear Prediction Coefficients LPC or Perceptual Linear Prediction PLP may improve their performances. Finally we have interest of ASPh implementation using other classifiers like Deep Learning Neural Networks.

6. References

[1] G. Almpanidis, C. Kotropoulos, “Phonemic segmentation using the generalised Gamma distribution and small sample Bayesian information criterion”, Speech Commun., vol. 50, no. 1, pp. 38-55, 2008.
https://doi.org/10.1016/j.specom.2007.06.005

[2] J. Yuan et al., “Automatic phonetic segmentation using boundary models”, Interspeech, pp. 2306-2310, 2013.

[3] H. Romsdorfer, B. Pfister, “Phonetic labeling and segmentation of mixedlingual prosody databases”, Interspeech, pp. 3281-3284, 2005.

[4] S. Hoffmann, B. Pfister, “Fully automatic segmentation for prosodic speech corpora”, Interspeech, pp. 1389-1392, 2010.

[5] M. S. Abdo, A. H. Kandil, S. A. Fawzy, “MFC Peak Based Segmentation for Continuous Arabic Audio Signal”, Middle East Conference on Biomedical Engineering (MECBME) , IEEE, pp- 224-227, Qatar, February 17-20, 2014.
https://doi.org/10.1109/mecbme.2014.6783245

[6] A. Abed, A. Amrouche, A.K. Delmadji, K. Boubakeur, G. Droua-Hamdani, “Segmentation Automatique des Signaux Sonores par HMM et RNA pour la Langue Arabe,” in *Proc. CISTEM 2016,* October 26-28, 2016, Marrakech, Morroco, 2016.

[7] HTK speech recognition toolkit : http://htk.eng.cam.ac.uk/, last visited, December 2016.

[8] C. Karin, A reference grammar of modern standard arabic, Cambridge university press, 2005.

[9] D. Cohen, Langue arabe, Encyclopedia Universalis, pp. 707-732, Paris, 1990.

[10] A. I. Alfozan, Assimilation in Classical Arabic ; A phonological study, Thèse de doctorat, Faculty of Arts of the University of Glasgow, Scotland, 1989.