The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 1-2, 2013
Doi: 10.7321/jscse.v3.n3.135 e-ISSN: 2251-7545

Arabic Language Learning Software based on ASR and TTS systems

Mourad Abbas
Phonetics and Speech Processing Lab.
crstdla
Algiers, Algeria
m_abbas04@yahoo.fr

***Abstract*—In this paper, we present an Arabic language learning software that we have implemented by incorporating both Automatic Speech Recognition (ASR) and Text to Speech (TTS) systems. Indeed, this educational software allows, in the first step, the non-Arabic speakers and/or learners of primary education to acquire the Arabic alphabet and some basic rules of standard Arabic; And helps users, in the second step, to practice reading. As the software requires the use of the two aforementioned systems, we will present firstly the experiments on an isolated word recognition system that we have achieved, and secondly we will give a description of the educational software.**

Keywords—Educational Software; Arabic language; ASR; TTS.

I. INTRODUCTION

The use of ASR systems and/or speech synthesis in software development for language learning has become very necessary, thanks to the considerable contribution offered by these systems [1,2,3]. The ultimate goal of this application is to make learning Arabic easier. Indeed, this can be achieved by exploring the alphabet which is considered as an essential step to learn any language. This software allows learners of primary education and non-Arabic speakers to learn the Arabic alphabet by listening with repetition, and gives them the opportunity to assess themselves by pronouncing any letter and compare it to the corresponding reference waveform. Hence, the learner can pronounce a phoneme which is immediately recognised by using the ASR module. This operation is important since it allows determining which letter has been pronounced by the learner. The signal corresponding to the pronounced letter (or the word) and the score displayed on the screen are the two parameters used by the learner for self-assessment. This software is not only limited to learn letters, but it deals with some specificities of the Arabic language such as "Madd" (long vowel), "Shadda" (gemination), as well as treating the teaching of reading arabic texts using the TTS system. In addition, it includes a learning space where the user may input a text and listen, each time he types a letter, the corresponding sound wave. On the other hand, the used ASR system is based on Hidden Markov Models, which are considered among the most frequently used tools in automatic speech recognition. Thus, we chose a well-defined corpus, namely: the standard Arabic alphabet, which is more suited to our application. We'll give more details describing the system in the following sections. The TTS system, meanwhile, contains two modules: a word processing module whose role is to convert the text into a phonetic string, and a signal processing module which generates the acoustic wave corresponding to this string chain.

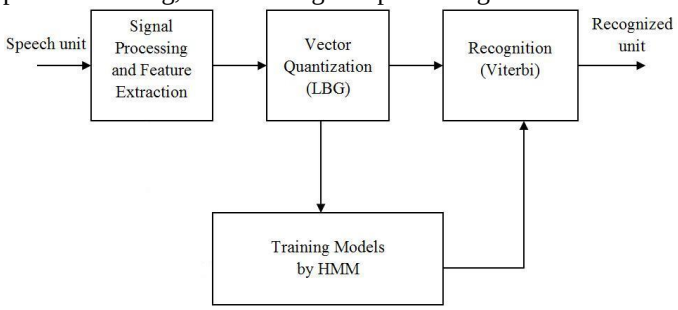


Fig. 1. Speech recognition phases.

We'll give more details describing the system in the following sections. Our TTS system, meanwhile, contains two modules: a word processing module whose role is to convert the text into a phonetic string, and a signal processing module which generates the acoustic wave corresponding to this string chain.

II. AN OVERVIEW ON THE ASR MODULE

The ASR module presents the core of the software since it allows the self-assessment of the learner after recognizing theuttered letters. The ASR module is not designed to recognize continuous speech, but to single words. For this, we chose the entire Arabic alphabet as isolated words to be recognized. Fig. 1 presents the general steps we followed to build this recognizer.

A. Speech Analysis

Speech recordings were made by using the Auros card AU21. The 28 letters of the Arabic alphabet were recorded, at a sampling frequency 10 kHz, eight times by a dozen speakers. This set of recorded words will be used in the phase of models training, explained in section II-B. In order to have a better representation of the speech signal, we used the weighted cepstral coefficients, their derivatives and second derivatives.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 1-2, 2013
Doi: 10.7321/jscse.v3.n3.135 e-ISSN: 2251-7545

The choice of the weighted cepstral coefficients and their derivatives is justified by their representation of spectral changes function of time and their robustness against variations of the speaker [4], [5]. The weighting of these coefficients is intended to address the problem of the inadequacy of speech signals caused by recording speech in different environments. We estimated the cepstral coefficients using the LPC coefficients which offer a much reduced computational time, in a very straightforward manner. Equations (1) and (2) show the relationship between these coefficients:

$$ln \quad \frac{1}{A_p(Z)} \quad = \sum_{n=1}^{\infty} C_q(n) Z^{-n} \qquad (1)$$

$$C_q(i) = -a_p(i) - \sum_{n=1}^{i-1} a_p(n) C(i-n) \qquad (2)$$

with $A_p(Z)$: represents the inverse filter.

C_q : cepstral coefficients.

a_p : linear prediction coefficients.

In order to improve the performance of the recognizer, the acoustic vector is composed of the static parameters (cepstral coefficients), completed by delta and delta-delta coefficients which are derivatives and second derivatives of cepstral coefficients. Delta parameters are calculated by using (3) :

$$\Delta_t = \frac{\sum_{\tau=1}^{\delta} \tau (C_{t+\tau} - C_{t-\tau})}{2 \sum_{\tau=1}^{\delta} \tau^2} \qquad (3)$$

From equation (3) we see clearly that Δ_t which corresponds to the delta coefficient at time *t* calculated by using the cepstral coefficients $C_{t-\tau}$ to $C_{t+\tau}$ with a delay δ. The number of recorded units is 2240. These units are represented by acoustic vectors which are composed of 8 weighted cepstral coefficients, their derivatives and their second derivatives. The amount of vectors will be used to build a codebook. This need only to be realized once, in a computationally intense training phase, by using the LBG algorithm [6], [7] which performance is shown in Table (I). The resulting codebook is then used to quantize general test vectors.

TABLE I. SNR (SIGNAL NOISE RATIO) VALUES FOR DIFFERENT SIZES OF THE CODEBOOK

Size of the codebook	256	128	64	32	16
SNR (dB)	4.31	3.78	2.99	2.17	1.66

B. *HMM based recognition*

The most widely used method of building acoustic models for speech units (phonemes or words) is the well-known statistical Hidden Markov Models [8], [9]. Each speech unit is represented by a HMM model. Therefore, since we have 28 alphabet units to be recognized, the number of models that will be trained is 28. A *Q*-state HMM is characterized by the following [11], [10]:

- A number of *Q* states $S = \{S_1, S_2, \ldots, S_Q\}$.
- A state transition matrix $A = \{a_{i,j}, 1 \le i,j \le Q\}$which represents a set of state transitions, a_{ij}, which specify the probability of making a transition from state *i* to state *j* at each frame.
- A state observation probability density, $B = \{b_j(x_t), 1 \le j \le Q\}$, where x_t is the acoustic feature vector at time *t*.
- An initial state distribution, $\pi = \{\pi_i, 1 \le i \le Q\}$.

The approach used in the design of our HMM based ASR is a maximum likelihood (ML). This philosophy asserts that a model is "good" if its parameters are adjusted to maximize the probability of generating the observation (training) sequences for which it is "responsible"[12]. It is essential to train a given HMM with multiple training utterances, in order to provide a more complete representation of the statistical variations likely to be present across utterances[12]. This is achieved by the Baum-Welsh algorithm which is iterated until a stable alignment of models and speech is obtained [10]. Recognition is achieved by using the Viterbi algorithm. Hence, we need to compute the probability that the acoustic vector sequence $X = \{x_1, x_2, \ldots, x_T\}$came from the word (or the sub-word) sequence $W = w_1, w_2, \ldots, w_M$. This calculation can be expressed as:

$$P_A(X|W) = P_A(\{x_1, x_2, \ldots, x_T\} | w_1, w_2, \ldots, w_M) \qquad (4)$$

C. *Recognition tests*

We conducted tests on the recognition system using the recordings representing the Arabic alphabet. Two-thirds of the Data have been used for models training, namely 1494 speech units. The remaining third is reserved for testing. We obtained a recognition rate equal to 98%.

III. DESCRIPTION OF THE EDUCATIONAL SOFTWARE

A. *Introduction*

The idea of designing educational software to learn a second language is consolidated by the emergence of ASR and TTS systems. Indeed, the most important motivation for building such software is that users may learn and assess themselves. As we mentioned above, the design of this software is mainly based on the use of an automatic speech recognition system that we described in Section II. The software allows learning the Arabic alphabet by listening with repetition. In addition, it gives the ability to a self assessment by comparing the pronounced letter to a prestored reference letter. The 28 reference letters are represented by 28 Markov models which have been obtained after training the acoustic models, that required recording of the letters by a dozen people many times. The role of the models is to capture the variability of the speech signal. The two waveform signals corresponding respectively to the pronounced letter and the reference letter are displayed. This facilitates the task of visual comparison to the learner. Self evaluation is provided by the visualization of the score that the learner has achieved. Indeed, this score is the probability expressed by (4). The learning method by using the Arabic learning software can be summarized in the three following steps:

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 1-2, 2013
Doi: 10.7321/jscse.v3.n3.135 e-ISSN: 2251-7545

- Learning the Arabic alphabet.
- Intensive learning of the Arabic alphabet.
- Learning reading in Arabic.

B. *Learning the Arabic Alphabet*

This part of the software represents the beginning phase which allows the learner to learn the Arabic alphabet by sound, image and video. Each letter is represented by a button that, once clicked, a voice corresponding to this letter is produced, offering him the opportunity to listen several times until learning how to articulate. In the same time, the speech waveform corresponding to the letter is displayed. The learner may use the property of displaying speech waveforms in the second phase of learning in order to compare with his own pronunciation. The software is endowed with a video[1] for showing better the pronunciation of the words' letters that the learner selects. Furthermore, a list of words which contain the pronounced letters is displayed. A simple click on the word enables the generation of the corresponding sound wave, (see Fig. 2).

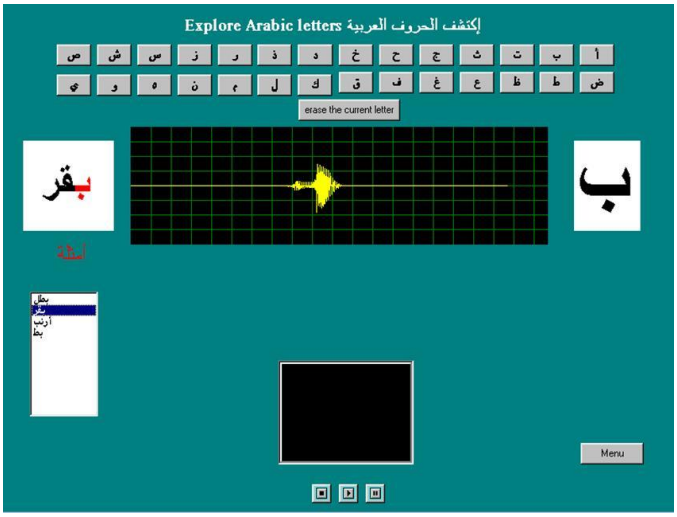
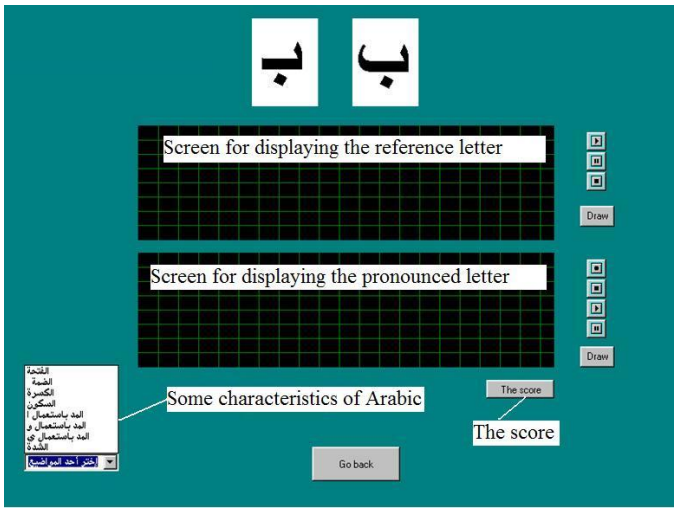


Fig. 2. The first step for learning the Arabic alphabet.

C. *Intensive Learning of the Arabic Alphabet*

In this phase, the users strengthen their knowledge by learning other characteristics of Arabic language as the "Madd"(long vowel) and "Shadda" (gemination). In addition, the software offers the learners the ability to record letters and words by their own voice and to make a self-assessment, by reading a score provided by the ASR system. In fact, the score is obtained by calculating the probability expressed by (4) using the Viterbi algorithm. Moreover, the displayed speech waveforms which correspond respectively to the letter pronounced by the learner and the letter of reference offer a visual comparison -Fig. 3-. In the interface shown in Fig. 3, we can get all the necessary information to learn a letter. Hence, the various forms in which the letter can be found in a text, its pronunciation, examples of words that contain the letter in question and the corresponding pronunciation, and finally the score that helps to give the self-assessment criterion. Additional features of Arabic language such as "Harakat" (vowels: a, u, i) are learned the same way mentioned above, i.e. by clicking on the "Features" button.

Fig. 3. Advanced learning of the Arabic alphabet by using the self-assessment criterion is one of the most important points followed in this step.



D. *Learning Reading in Arabic*

In order to learn how to read in Arabic, a collection of texts is available. Learners may choose a text and start listening. They may also listen to a word or a sub-word as much as they want, by selecting it with the mouse in order to be converted by the TTS system. Moreover, the software allows users to learn the fondamental rules of reading as well as to avoid some frequent errors like the following:

- Elision: eg. if the word (إستقبال) is preceded by the preposition (من), the Hamza (ء) must be removed from the letter (إ), and the word must be red (من استقبال) instead of (من إستقبال).
- Gemination: if the learner faces problems to pronounce a geminated letter correctly, a Help button is used to give more details about gemination, by referring to the corresponding geminated letter, under the heading of gemination presented among other characteristics in the list box -Fig. 3-.
- Movement (vowels) and pause (Sukun)[2]: in Arabic, vowels should not be used in the end of the uttered speech, even they exist in the written form. Instead of that, the sukun will be used in this case. This is a very basic rule which is ignored even by the instructors of the Arabic language. If this mistake is committed by the user, i.e. he prounouces the vowel in the end of its reading, it will be corrected immediately by sending a sound message. In fact, the detection of the vowel which is a voiced speech is realized easily by the ASR system.

Moreover, in this phase of learning, we have intergrated a space for editing texts. Hence, the learner finds writing easier

[1] The visualization of the person articulating the word is optional.

[2] Commonly known as Haraka (الحركة) and Sukun (السكون) by the arabian Grammarians. More details about these two notions can be found in the paper of the Algerian linguist Abderahmane Hadj-Salah, intitled La notion de syllabe et la theorie cinetico-impulsionelle des phoneticiens arabes. Algerian Journal of Linguistics (1971).

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 1-2, 2013
Doi: 10.7321/jscse.v3.n3.135 e-ISSN: 2251-7545

since he listens each time the letters as soon as he hits the keybord. This is achieved by using the TTS system.

IV. CONCLUSION

In this paper we presented our software for learning the Arabic alphabet for non-Arabic speakers and/or learners of primary education. Firstly, we presented the steps that we followed to achieve an ASR system for isolated words recognition based on HMM. We believed that the recognition rate obtained (98%) is significant and useful for the integration of the recognizer in the software. Secondly, we gave a description of the educational software which is strengthened by the ASR system that provides the self-assessment criterion which conducts to a self-managed learning. The general design of the software gives users access to learn the Arabic alphabet in the first two steps and allows them to acquire the rules of reading in Arabic in the third step. In this first version of the software, we have not addressed all the rules to learn reading. However, we have shown and used the most important ones to better learn prununciation and spelling for a better reading acquisition.

REFERENCES

[1] R. Hincks, "Speech recognition for language teaching and evaluating: A study of existing commercial products," Proc. ICSLP, Denver, 2002, pp. 773–776.

[2] Z. Liu, "Research on pronunciation scoring technology in language training system," Proc. Int. Conf. on e-Education, Entertainment and e-Management, Bali, 2011, pp. 212–215.

[3] R. Hincks, "Speech synthesis for teaching lexical stress," TMH-QPSR 44 (2002) pp. 153–156.

[4] S.D. Furui, "Speaker-Independent Isolated Word Recognition using Dynamic Features of Speech Spectrum," IEEE Trans. Acoustics, Speech, and Signal Processing 34(1) (1986) pp. 52–59.

[5] J.W. Picone, "Signal Modeling Techniques in Speech Recognition," Proc. IEEE 81(9) (1993) pp. 1215–1247.

[6] A. Gersho and R. Gray, Vector quantization and signal compression, Kluwer Academic Publishers, Boston / Dodrecht / London, 1992.

[7] Y. Linde, A. Buzo and R. Gray, " An algorithm for vector quantizer design," IEEE Trans. Comm. 28 (1980) 84–95.

[8] J. D. Ferguson, Hidden Markov Analysis: An Introduction, Hidden Markov Models for Speech, Princeton: Institute for Defense Analyses, 1980.

[9] S. E. Levinson, L. R. Rabiner and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," Bell System Technical Journal 62(4) (1983) 1035–1074.

[10] L. R. Rabiner and R. W. Schafer, Introduction to Digital Speech Processing 1:1-2, now Publishers Inc., USA, 2007.

[11] M. Abbas and M. Debyeche, "Reconnaissance Automatique des Phonemes Arabes dans la Parole Continue," Proc. 7th Magrebian Conference on Computer Sciences, Annaba, Algeria, 2002, pp. 79-87.

[12] J. R. Deller, J. H. L. Hansen and J. G. Proakis, Discrete-Time Processing of Speech Signals IEEE press, New York, 2000.

[13] X. D. Huang, Y. Ariki and M. A. Jack, Hidden Markov models for speech recognition, Edinburgh University Press, Edinburgh, 1990.

J. P. Haton, Reconnaissance Automatique de la Parole, Bordas, Paris, 1991.