Design and Implementation of a Diacritic Arabic Text-To-Speech System

Aissa Amrouche, Leila Falek, and Hocine Teffahi
Electronics and Computer Science Faculty, University of Sciences and Technology Houari Boumediene, Algeria

Abstract: *The absence of the diacritical marks from the modern Arabic text generates a significant increase of the ambiguity in the Arabic text, which can cause confusion in the pronunciation of a written word. Despite the fact that the reader with a certain level of Arabic knowledge can easily recover the missing diacritics by: using the words context, the morphology and the syntax knowledge of the Arabic language. This paper describes a design and implementation of a Text-To-Speech system for a diacritic Arabic text. The goal of this project is to obtain a set of high quality speech synthesizer based on unit selection using a bi-grams model taking into account the particularities of the language. It takes a diacritic Arabic text as input and produces corresponding speech; the output is available as male voice. The evaluation of our TTS system is based on subjective and objective tests. The final evaluation of GArabic TTS system, regarding the intelligibility, naturalness aspects (listening) and the quality (PESQ) is jugged successful.*

Keywords: *Diacritics, Arabic Language, Diacritization, Text-To-Speech, Speech synthesis, Unit selection, Bi-grams model.*

Received January 8, 2015; accepted April 23, 2015

1. Introduction

Text-to-Speech systems are used to convert words from a computer document (e.g. word processor document, web page) into audible speech spoken through the computer speaker. A text-to-speech system must be : able to read any text, Intelligible and natural sounding. The benefits of speech synthesis have been many, including: oral access to any written information such as fax, e-mail, textual databases; better hearing aids more simultaneous telephone conversations on the same cable; talking machines for vocally impaired or deaf people and better aids for speech therapy.

The Arabic Text-to-Speech systems are still in its infancy, compared to other languages such as English, there are currently several commercially available Arabic TTS systems such as ARABTALK, BrightSpeech, ElanSpeech and Sakhr TTS. However the Arabic language is spoken by almost 300 million people in the world wide and 22 countries as well. The intended pronunciation of a written word cannot be completely determined by its standard orthographic representation; rather, a set of special diacritics is needed to indicate the intended pronunciation. Different diacritics over for the same spelling produce different words with may be different meanings. Arabic writing system consists of 36 letter forms which represent the Arabic consonants. These are: ح , خ , د , ذ , ر , ز , س , ش , ج , ث , ت , ب , ة , ى , ء , ؤ , ئ , إ , أ , آ ,ا و , ه , ن , م , ل , ك , ق , ف , غ , ع , ظ , ط , ض , ص , and ي . Each Arabic letter represents a single consonant with some exceptions: ؤ , ئ , إ , أ and ء represent the glottal stop, but are written in different forms depending on the consonant position in the word and its adjacent phonemes. ا symbolizes the glottal stop or the long low vowel depending on its position in the word and sentence. و and ي are long vowels when preceded by a vowel of its nature dhammah and kasrah respectively, and they are consonant otherwise [1]. In addition there are 8 diacritics marks may be placed above or below a letter Table 1.

Table 1. Arabic diacritics with their International Phonetic Alphabet representations (IPA), definitions and samples with the voiced alveolar letter

Diacritic	IPA	Definition	Sample
◌َ	A	fathah (low short vowel)	زَ
◌ُ	U	dhammah (high back rounded vowel)	زُ
◌ِ	I	kasrah(high front vowel)	زِ
◌ّ	:	shaddah (geminate: consonant is doubled in duration)	زّ
◌ْ	Ø	sukoon (the letter is not diacritized nor geminated)	زْ
◌ً	An	tanween fathah (low vowel + alveolar nasal)	زً
◌ٌ	Un	tanween dhammah (high back rounded vowel + alveolar nasal)	زٌ
◌ٍ	In	tanween kasrah (high front vowel + alveolar nasal)	زٍ

The first three diacritics represent the Arabic short vowels, and the last three are the tanween that occur only in word final position. Almost all modern Arabic texts are written using the consonant symbols only. A word such as "علم" when diacritized can be: "عَلَم" flag, "عِلْم" science, "عُلِمَ" it was known, "عَلِمَ" he knew, "عَلَّمَ" he taught or "عُلِّمَ" he was taught. Arabic readers infer the appropriate diacritics based on the linguistic knowledge and the context. However in the case of a Text-To-Speech or automatic translation system,

Arabic letters need to be diacritized, otherwise, the system will not be able to know which word to select. There are general rules for diacritizating Arabic text [7]. Texts without diacritics present an obstacle for non-native learners of the Arabic language and those with learning difficulties. Similarly, the performance limits of several applications of natural language processing for Arabic (NLPA) such as parsers and Treebank are in part a result of the absence of diacritics in Arabic texts [11, 20]. Indeed, unlike European languages where it is easy to identify oral phonemes corresponding to texts (Text-To-Speech), it is imperative for Arabic texts to retrieve the diacritics before researching the correspondent oral phonemes [18]. On the other hand, some research has underlined the importance of using texts with diacritics to increase the efficiency of speech recognition [12]. In this article, we present a Text-To-Speech system called "GArabic TTS" for a diacritic Arabic text. The system is based on unit selection method in a large database. The design of a database "corpus" and the various required steps to implement the developed method have been described.

2. The Arabic Database Construction

The database must be large enough to contain all kinds of phonetic sequences that can appear in different linguistic contexts, so the result speech segments are available in different prosodic forms. In this study, for simplicity, we chose 50 sentences from the Rosetta Stone software [9] (it is a language learning computer-assisted (CALL) software published exclusively by Rosetta Stone) spoken by a male speaker. The total vocabulary is about 6000 units. We gave the name GArabic (Generic Arabic) to our database which constitute the corpus "GArabic_corpus". The database must be prepared for the selection method has all the information necessary for its operation.

It consists of:

- A record file «. wav» which consists of all sentences in the corpus. Each sentence is named by a number such as: 1.wav, 2.wav, …, 50.wav.
- A file «.m» which contains the orthographic transcription, parts-of-speech, and the phonetic transcription of the database (see Figure 1).
- A file «.seg» of all phonetic transcriptions is associated with each phone in the sentence as well as their limitations. Each sentence is named by a number such as: 1.seg, 2.seg, …, 50.seg.

The segmentation is handmade into phonemes by the Praat software (Paul Boersma and David Weenink, 2014). In order to check the resulting segmentation and correct if when necessary, we have used the WavSurfer tool (Jonas Beskow and Kare Sjolander, 2011).

```
GArabic_corpus = {
اَلرَّجُلُ وَ اَلْوَلَدُ يَجْلِسَانِ عَلَى اَلدَّرَّاجَةِ لَكِنَّهُمَا لاَ يَرْكَبَانِ اَلدَّرَّاجَةَ %
 'اَلرَّجُلُ'    'noun'          'A_Ra_ju_lu'
 'وَ'            'coordinator'   'wa'
 'اَلْوَلَدُ'    'noun'          'Al_wa_ladu'
 'يَجْلِسَانِ'   'verb'          'ya_jlisa:ni'
 'اَلدَّرَّاجَةِ' 'noun'         'A_Da_Ra:ja_ta_'
 'عَلَى'         'preposition'   'X_a_l_a'
 'وَ'            'coordinator'   'wa'
 'لَكِنَّهُمَا'  'verb'          'lakiNa_hu_ma'
 'لاَ'           'prefixes'      'la_'
 'يَرْكَبَانِ'   'verb'          'yar_kaba_ni'
 'اَلدَّرَّاجَةَ' 'noun'         'A_Da_Ra:ja_t_a'
 '.'             'punctuation'   '_'
 ……..
```

Figure 1. A sample of GArabic corpus file.

From this segmented speech corpus, we have built a speech unit database, in which we have stored, for each available unit, the minimum information needed to compute its match to a given phonemic target (Figure 2). They are:

- Its phoneme, previous phoneme and next phoneme,
- The index of the part-of-speech of the current word,
- The index of the current prosodic phrase (within the current sentence),
- The number of prosodic phrases on the right (until the end of the sentence),
- The number of words on the right (until the end of the current prosodic phrase),
- The index of the current word (within the current prosodic phrase),
- The index of the sentence containing the phoneme (related wav file names are given by this index),
- And the start/end sample for the current phoneme in the related wav file.

```
>>corpus=corpus_to_speech_corpus(GArabic_corpus)
>>corpus=corpus(1:3,:)
        '_#An1015'   [1]  [0]      [1318]
        'A#Rn1015'   [1]  [1318]   [2216]
        'RAan1015'   [1]  [2216]   [3085]
```

Figure 2. Presentation of three units in the speech unit database.

The second unit information must be understood as follows: the phoneme A is preceded by nothing (#) and followed by R in a noun (n) in the sentence [1.wav] and the sample is from [1318] to [3085].

3. The Implementation of the method

The different implementation method stages are

3.1. Lexicon Creation

The lexicon is made from the corpus GArabic using the «Garabic_load_corpus.m» function. This script contains the corpus sentences, the part-of-speech and the phonetic transcription of each word (Figure 3).

```
>> corpus=GArabic_corpus(1:3,:)
>> corpus =

    'اَلرَّجُلُ'   'noun'          'A_Ra_ju_lu'
    'وَ'           'coordinator'   'wa'
    'اَلْوَلَدُ'   'noun'          'Al_wa_ladu'
```

Figure 3. The first three words representation.

3.2. Corpus preprocessing

Our GArabic corpus although is large, is distinguished by its simplicity: it contains no numbers, acronyms, or complex proper names. Furthermore, the synthesized sentences have to contain no spelling mistake. Thus, the only task left to do is to decompose the input text into states set (words and punctuation). The implementation of this strategy in Matlab is made by the function «decompose» using the «strtok» function Figure 4.

```
>> Phrase=' اَلرَّجُلُ وَ اَلْوَلَدُ يَجْلِسَانِ عَلَى اَلدَّرَّاجَةِ لَكِنَّهُمَا لاَ يَرْكَبَانِ الدَّرَّاجَةَ'
>> sentence=decompose(phrase)
   sentence =

    'اَلرَّجُلُ'
    'وَ'
    'اَلْوَلَدُ'
    'يَجْلِسَانِ'
    'عَلَى'
    'اَلدَّرَّاجَةِ'
    'لَكِنَّهُمَا'
    'لاَ'
    'يَرْكَبَانِ'
    'اَلدَّرَّاجَةَ'
```

Figure 4. The decomposition result of first sentence.

3.3. Phonological analysis of the GArabic Corpus

We previously reported the advantages of the GArabic corpus and we showed that the database strength resides in the hierarchical composition of its files. Do not forget, of course, the possibilities offered by the phonetic transcriptions files. Thereby, the phonological analysis of GArabic corpus consists in presenting a list of words in the corpus associated with their parts-of-speech. Thus, we cover all the directories and files «.wav», «.seg» to create this list. The function «corpus_to_list.m» creates the matrix «word_list» and presents words with their corresponding phonetic transcriptions Figure 5.

```
>> [word_list _of_S]=corpus_to_directory(GArabic_corpus);
                word_list =
                ','          {1x1 cell}
                '.'          {1x1 cell}
                'اَلرَّجُلُ'   {1x1 cell}
                'اَلْأَبْيَضِ'  {1x2 cell}
                'هُنَاكَ'     {1x1 cell}
            'وَ'         {1x1 cell}
        'اَلْوَلَدُ'     {1x1 cell}
        'يَجْلِسَانِ'    {1x1 cell}
        'عَلَى'          {1x1 cell}
```

Figure 5. The word list.

For example, in word_list, we got two parts-of-speech of the word 'اَلْأَبْيَضِ' such as:

```
>> w=word_list{4,2}
   w=    'adjective'   'noun'
```

The same function «corpus_to_ directory.m» allows us to have all the parts-of-speech Figure 6.

```
>> P_of_S
   P_of_S =
        'adjective'
        'adverb'
        'coordinator'
        'demonstrative'
        'noun'
        'verbe'
        'preposition'
        'pronoun'
        'propername'
        'punctuation'
        'subordinator'
```

Figure 6. The parts-of-speech.

Then we created a Matlab function «directory_search.m», which allows us to find the parts-of-speech for a given word such as:

```
>> possible_word=directory_search('اَلْأَبْيَضِ',word_list)
   possible_word =  'adjective'   'noun'
```

The function «tts_morph_using_directory» is used to find all the possible parts-of-speech corresponding to words which constitute the sentence to be synthesized Figure 7.

```
   >>possible_tags=tts_morph_using_directory({'أَمَامَ';'اَلْأَبْيَضِ'}, word_list)
   possible_tags =
                {1x1 cell}   {1x2 cell}
   >>possible_tags{:,:}
   ans =
        'preposition'
   ans =
       'adjective'   'noun'
```

Figure 7. The parts-of-speech possibilities corresponding to a given word.

3.4. Unit Pre-Selection

At this stage, we are able to associate for each word the part-of-speech that matches it. This implies that we must make a selection on all parts of speech proposed by the previous modules. The standard strategy was to use the n-grams model [4, 6, 8] that we will present in the next paragraph. The model n-grams appeared in the speech recognition context to estimate a sequence of words $W_1,\ldots, W_N$ in a given language.

In the pre-processing context for a given sentence $W=(w_1, w_2,\ldots,w_N)$, we find the best tag sequence $\hat{T}$ over all possible sequence $T=(t_1,t_2,\ldots,t_N)$, carried out of all part-of-speech tags $\{c_1,c_2,\ldots,c_M\}$:

$$\hat{T} = \arg\max_{T} P(T/W) \quad (1)$$

By Bayes's rule, this is equivalent to find:

$$\hat{T} = \arg\max_{T} \frac{P(T,W)}{P(W)} = \arg\max_{T} \frac{P(W/T)}{P(W)} \quad (2)$$

The denominator of Equation (6) is independent of T, so we can ignore it in the search of $\hat{T}$. The N-grams model for pre-processing has the following approximations:

- The probability of a word given the past mostly depends on its tag.
- The probability of tag given the past mostly depends on the last N-1 tags.

As result:

$$P(T/W) = P(w_1,w_2,...,w_N/t_1,t_2,...,t_N)$$
$$= P(w_1/t_1,t_2,...,t_N)P(w_2/w_1,t_1,t_2,...,t_N)...P(w_N/w_1,...,w_{N-1},t_1,t_2,...,t_N) \quad (3)$$
$$\approx \prod_{i=1}^{N} P(w_i/t_i)$$

$$P(T) = P(t_1,t_2,...,t_N)$$
$$= P(t_1)P(t_2/t_1),...,P(t_N/t_1,t_2,...,t_{N-1}) \quad (4)$$
$$\approx \prod_{i=1}^{N} P(t_i/t_{i-1},t_{i-2},...,t_{i-N+1})$$

It is clear that we can model the problem by a finite state automaton. This automaton shows a bi-grams model, where n = 1 [4, 6].

The bi-grams model considered is represented by associated states with possible parts of speech (one state per part-of-speech).

At each transition we associate a probability $p(c_i \mid c_j)$ that represents the probability of a word with the category c_j will be followed by a word with the c_i category.

The emission probabilities $p(wi \mid cj)$ represent the probability that the category c_j correspond to the word w_i. An example of bi-grams model is given in Figure 8:

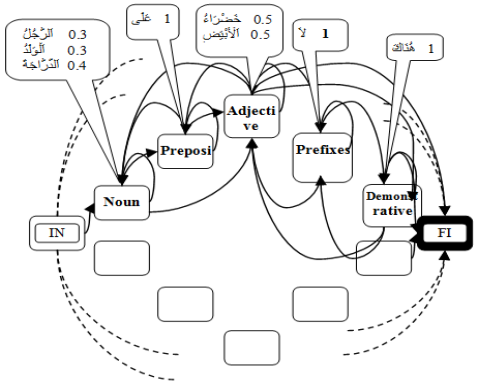

Figure 8. A possible Bi-grams automaton for GArabic Corpus (all states are supposed to be fully connected: only a few connections are shown).

The calculation of the emission probability is simple. In fact, this probability is approximately given by the number of times w_i apears as c_i divided by the total number of words with parts-of-speech c_i:

$$P(w_i \mid c_j) = \frac{\#(w_i,c_j)}{\#(c_j)} \quad (5)$$

Similarly, the transitional probabilities between two categories c_j and c_i represents the number of times c_i appears after c_j divided by the total number of words with part-of-speech category c_j.

$$P(c_i \mid c_j) = \frac{\#(c_i,c_j)}{\#(c_j)} \quad (6)$$

The calculation of these probabilities was provided by the Matlab function «corpus_to_bigrams.m», which returns the values of emission and transition probabilities Figure 9.

```
[emission_probs, transition_probs]=corpus_to_bigrams
(GArabic_corpus);
>>emission_probs(:,5)        transition_probs(:,1)
   ans = 1                      ans = 0.6667
         0                            0
         0                            0
         0                            0.3333
         0                            0
```

Figure 9. The emission and transition probabilities.

For example, the column 5 of "emission_probs" has a non-zero value, which explains the fact that the fifth category of part-of-speech P_of_S (prefixes) may appear as one GArabic word «'لِ'» with its emission probability equal to 1.

Similarly, the column 1 of «transition_probs» has two non-zero values. This explains the fact that the first category of part-of-speech P_of_S (adjective) is followed by an adjective and a preposition in the training corpus, with probability 0.6667, 0.3333 respectively.

Though, one can never be sure to cover all possible cases in a corpus, however, large it is. People typically address this problem by changing zeros into small non-zero values, which will tend to restrain the algorithm from choosing very unlucky paths, while avoiding the assumption of strict null probabilities. We adopt for this, the strategy adopted by Dutoit [6] that replace the null values with 1e-8.

3.5. Unit selection

Once the probabilities are estimated, it remains to find the best sequence with the highest probability. By analogy with the cost optimization procedure [2, 3, 5, 10, 13, 14, 15, 16, 17, 19], we can estimate the target cost by the inverse of the transmission probability and the concatenation cost by the inverse of the transitional probability. Thus, finding the best sequence minimizing the costs is to find the sequence maximizing the sum of the probabilities. This corresponds to find the best path in a lattice. As a matter of fact, while Figure 8 shows a Bi-grams automaton for all possible sentences of GArabic corpus, the automaton reduces to a lattice for a given sentence Figure 10.

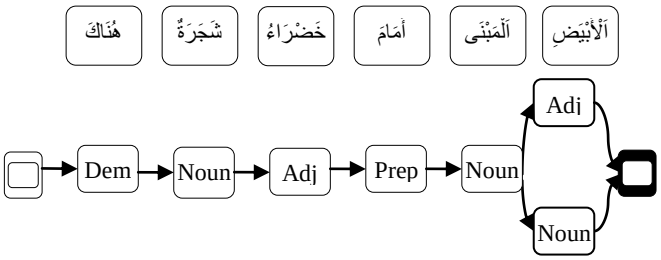

Figure 10. An example of a lattice Bi-grams for a simple GArabic sentence

To get the best sequence, we used the two functions «lattice_get_paths.m» and «tts_tag_using_bigrams.m» that we have applied to the bi-grams model Figure 11.

```
sentence={'هُنَاكَ';'شَجَرَةٌ';'خَضْرَاءُ';'أَمَامَ';'ٱلْمَبْنَى';'ٱلْأَبْيَضِ' };
possible_tags=tts_morph_using_directory(sentence,word_list);
tags=tts_tag_using_bigrams(emission_probs,transition_probs,wo
rd_list,P_of_S,sentence,possible_tags);
    tags =
        'demonstrative'
        'noun'
        'adjective'
        'preposition'
        'noun'
        'adjective'
```

Figure 11. The best sequence for a given sentence.

3.6. Speech synthesis

Once the units are selected from the speech corpus, we reconstruct the signal by concatenation. The algorithm used here is based on TD-PSOLA method [20].

4. Results And Discussions

An interactive graphical user interface has been designed and implemented under MATLAB environment, which allows the user an easy use of our synthesis system. In order to evaluate the speech of the developed system regarding the intelligibility and the naturalness aspects two types of tests were applied.

The first test which measures the intelligibility is divided in two-subtest. The intelligibility testing is performed using subjective test. In subjective tests, human listeners hear and rank the quality of processed voice files according to a certain scale. The most common scale is called MOS (Mean Opinion Score) and is composed of five scores of subjective quality, 1-Bad, 2-Poor, 3- Fair, 4-Good, 5-Excellent. The MOS score of a certain vocoder is the average of all the ranks voted by different listeners of the different voice file used in the experiment.

We randomly selected a group of people to evaluate the developed system. The participants are 50 with the age group of 18-50 years, different professions and knowledge of the Arabic language in order to get a good assessment purposes. The subjects listened the sentences using headphones.

In the first part, each participant hears and ranks on an answer sheet (four choices) which sentence is listened for (Test 1A). In the second part, ten sets of sentences are chosen; each one has four sentences, the sentences differ only in a single consonant on its words for the same set. The listeners are asked to mark on the answer sheet which number corresponds to the written sentence (Test 2A). The second test (quality), we evaluated the GArabic TTS system with MOS and PESQ tests. The PESQ is an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. It compares the original signal with the corresponding (degraded) synthesis signal.

After collecting all listeners' response, we calculated the average values. The final averages of the test results for the MOS and PESQ are 3.909, 2.915 respectively. The experimental results are summarized in Figure 12.

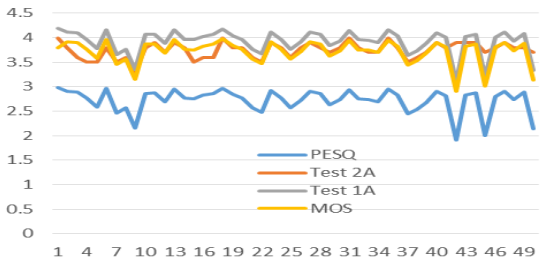

Figure 12. The assessement results test.

By analyzing the results, in the test 1A 92% of sentences are correctly checked. The MOS averge given by the listeners is 4.034, which imply that the participants can hear what is being said and recognize the majority of sentences. For the test 2A 88% of sentences are correctly checked and the averige of MOS test is 3.784. It is well known that when the concatenated speech is produced from the extracted diphones the operation tends to produce audible mismatches. The discontinuity problem arise in its clearest form because of the large co-articulatory effects that exists between adjacent units also the quality with some consonants may vary considerably and the controlling of pitch and duration may be in some cases difficult. Putting all that into consideration, the degraded naturalness of the concatenated word that has appeared in the results is reasonable. Acording to the quality and intelligibility results the GArabic TTS Synthesizer System is successful.

5. Conclusions

In this work, a speech synthesis system called GArabic TTS was implemented and tested under Matlab environment for the Arabic language. The unit database has been created from the segmented speech corpus, in which we have stored for each available unit, the minimum information needed to compute its match to a given phonemic target. The Bi-grams model has been used for unit selection. The results of

perceptual evaluation test indicate that the intelligibility and naturalness aspects are successful; all participants are satisfying about the quality of GArabic TTS. We can see this from the listening tests (MOS) and objective evaluation to compare quality (PESQ) by comparing the original and synthetic speech.

References

[1] Bebah M., Amine C., Azzeddine M., and Abdelhak L., "Hybrid Approaches For Automatic Vowelization of Arabic Texts," *International Journal on Natural Language Computing*, vol. 3, no. 4, pp. 53-71, 2014.

[2] Breen A. and Jackson P., "Non-Uniform Unit Selection and the Similarity Metric within BT's LAUREATE TTS System," *in Proceeding of 3rd ESCA International Speech Synthesis Workshop*, 1998.

[3] Charpentier F. and Stella M., "Diphones Synthesis Using an Overlap-Add Technique For Speech Waveforms Concatenation," *in Proceeding of International Conference on Acoustics, Speech, and Signal Processing*, Tokyo, pp. 2015-2018, 1986.

[4] Chen S. and Goodman J., "An Empirical Study of Smoothing Techniques For Language Modeling," *in Proceeding of the 34th Annual Meeting on Association for Computational Linguistics*, California, pp. 310-318, 1996.

[5] Donovan R. and Eide E., "The IBM Trainable Speech Synthesis System," *in Proceeding of 5th International Conference on Spoken Language Processing*, Sydney, pp. 1703-1706, 1998.

[6] Dutoit T. and Cernăk M., "TTSBOX: A Matlab Toolbox for Teaching Text-To-Speech Synthesis," *in Proceeding of International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, pp. 537-540, 2005.

[7] Duwairi R., "Arabic Text Categorization," *The International Arab Journal of Information Technology*, vol. 4, no. 2, pp. 125-131, 2007.

[8] Elberrichi Z. and Abidi K., "Arabic Text Categorization: a Comparative Study of Different Representation Modes," *The International Arab Journal of Information Technology*, vol. 9, no. 5, pp. 465-470, 2012.

[9] Language learning, Rosetta stone http://www.rosettastone.com , Last Visited 2015.

[10] Lee M., Lopresti D.P., and Olive J.P., "A Text-to-Speech Platform for Variable Length Optimal Unit Searching Using Perceptual Cost Functions," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 347-356, 2001.

[11] Maamouri M., Bies A., and Kulick S., "Diacritization; A Challenge To Arabic Tree Bank Annotation And Parsing," *in Proceeding of the British Computer Society Arabic NLP/MT Conference*, England, pp. 35-47, 2006.

[12] Messaoudi A., Lori L., and Gauvain J-L., "The Limsi rt04 b Arabic System," *in Proceeding Fall 2004 Rich Transcription Workshop*, Palisades, 2004.

[13] Nomura T., Mizuno H., and Sato H., "Speech Synthesis by Optimum Concatenation of Phoneme Segments," *The ESCA Workshop on Speech Synthesis*, Autrans, pp. 39-42, 1991.

[14] Pantazis Y., Stylianou Y., and Klabbers E., "Discontinuity Detection in Concatenated Speech Synthesis Based on Nonlinear Speech Analysis," *in Proceeding of 9th European Conference on Speech Communication and Technology*, Lisbon, pp. 1-4, 2005.

[15] Peng H., Zhao Y., and Chu M., "Perceptually Optimizing the Cost Function for Unit Selection in TTS System With one Single Run of MOS Evaluation," *in Proceeding of 7th International Conference Spoken Language Processing*, Colorado, pp. 2613-2616, 2002.

[16] Prudon R. and Alessandro C., "A Selection/Concatenation Test-to-Speech System: Databases Development, System Design, Comparative Evaluation," *4th ISCA Tutorial and Research Workshop on Speech Synthesis*, Perthshire, pp. 138-143, 2001.

[17] Toda T., Kawai H., Tsuzaki M., and Shikano K., "Unit Selection for Japanese Speech Synthesis Based on Both Phoneme Unit and Diphone Unit," *in Proceeding of International Conference on Acoustics, Speech, and Signal Processing*, Orlando, pp. 465-468, 2002.

[18] Vergyri D. and Kirchhoff K., "Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition," *in Proceeding of Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Geneva, pp. 66-73, 2004.

[19] Yi J. and Glass J., "Information-Theoretic Criteria for Unit Selection Synthesis," *in Proceeding of the 7th International Conference on Spoken Language Processing*, Colorado, pp. 2617-2620, 2002.

[20] Zitouni I., Sorensen J., and Sarikaya R., "Maximum entropy based restoration of arabic diacritics," *in Proceeding of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics Workshop on Computational Approaches to Semitic Languages*, Sydney, pp. 577-584, 2006.

Aissa Amrouche is a Phd student at Electronics and Computer Science Faculty USTHB Algeria. A Researcher at Scientific and Technical Research Center for the Development of the Arabic Language. He received his Magister's degree from Computer Science Faculty USTHB, Algeria. His main interests include Arabic language processing and speech synthesis.

Leila Falek Electronics Doctor. Director of Research. Speech communication and signal processing laboratory, Electronics and Computer Science Faculty, Telecommunications department, USTHB, Algiers.

Hocine Teffahi Electronics Professor, Director of Research, Speech communication and signal processing laboratory, Electronics and Computer Science Faculty, Telecommunications department USTHB, Algiers