Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 34 (2015) 43–60

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Text-to-speech synthesis system with Arabic diacritic recognition system☆

Ilyes Rebai*, Yassine BenAyed

MIRACL: Multimedia InfoRmation System and Advanced Computing Laboratory, University of Computer Science and Multimedia, Tunis Street km. 10, Technopole, Sfax, Tunisia

Received 26 January 2014; received in revised form 5 April 2015; accepted 17 April 2015
Available online 27 April 2015

Abstract

Text-to-speech synthesis system has been widely studied for many languages. However, speech synthesis for Arabic language has not sufficient progresses and it is still in its first stage. Statistical parametric synthesis based on hidden Markov models was the most commonly applied approach for Arabic language. Recently, synthesized speech quality based on deep neural networks was found as intelligible as human voice. This paper describes a Text-To-Speech (TTS) synthesis system for modern standard Arabic language based on statistical parametric approach and Mel-cepstral coefficients. Deep neural networks achieved state-of-the-art performance in a wide range of tasks, including speech synthesis. Our TTS system includes a diacritization system which is very important for Arabic TTS application. Our diacritization system is also based on deep neural networks. In addition to the use deep techniques, different methods were also proposed to model the acoustic parameters in order to address the problem of acoustic models accuracy. They are based on linguistic and acoustic characteristics (e.g. letter position based diacritization system, unit types based synthesis system, diacritic marks based synthesis system) and based on deep learning techniques (stacked generalization techniques). Experimental results show that our diacritization system can generate a diacritized text with high accuracy. As regards the speech synthesis system, the experimental results and subjective evaluation show that our proposed method for synthesis system can generate intelligible and natural speech.

Keywords: Text-to-speech synthesis; Statistical parametric; Deep neural networks; Natural language processing; Diacritization system

1. Introduction

Text-To-Speech (TTS) system is one of the most important technologies due to the expanding field of applications, such as: multimedia, telecommunication and aids for handicaps. People that speak Arabic as their native language are more than 442 million around the world. Five million Arabic people are blind around the world (Zaki et al., 2010).

☆ This paper has been recommended for acceptance by Mark J.F. Gales.

* Corresponding author. Tel.: +216 21447305.
E-mail address: rebai_ilyes@hotmail.fr (I. Rebai).

http://dx.doi.org/10.1016/j.csl.2015.04.002
0885-2308/© 2015 Elsevier Ltd. All rights reserved.

Hence, Arabic TTS with intelligible and natural speech quality is required. However, the field of speech synthesis for Arabic language has not sufficient progresses and it is still in its first stage. This could be explained by the fact that:

- One of the problems facing computer processing of the Arabic text is the absence of the diacritic marks in the modern text (Elshafei et al., 2006). These marks are used to identify the right pronunciation of the text.
- It is difficult to obtain an Arabic speech database for speech synthesis task. The solution consists of developing a specific speech database (Chouireb and Guerti, 2008; Hamad and Hussain, 2011).
- Linguistic researches for Arabic language are limited.

While native Arabic readers can determine the appropriate vocalization of the text with minimal difficulty, computer processing of Arabic text is often obstructed by the lack of diacritic signs. For instance, a given Arabic TTS would not generate speech from undiacritized text because there are different pronunciations of the same undiacritized word. To make the problem easier for English readers, if the words “read, red, ride, rude” are written without vowels, they will be the same word “rd” and it will be impossible to determine the correct meaning and pronunciation of this word. Therefore, the Arabic diacritization system is very important for an Arabic TTS application. This system recognizes the missing diacritics of the input text. The problem of diacritic sign restoration can be solved by three approaches: standard Arabic dictionaries, rule-based approach and machine learning approach.

Speech synthesis system is the process of generation of speech, as output, from text, as input. The two popular approaches are *concatenative speech synthesis* (known as corpus-based approach) (Hunt and Black, 1996) and *statistical parametric speech synthesis* (called also knowledge-based approach) (Black et al., 2007). In the first approach, desired speech is produced by selecting and concatenating required segments from pre-recorded speech by human. Many systems use a corpus of fixed length units, typically phonemes or diphones. Other concatenative systems use more varied, non-uniform units speech segments database. For instance, in (Hamad and Hussain, 2011), the authors developed an Arabic TTS based on allophone and diphone concatenation method. This variety of speech segments allows the generation of more natural speech. The highest speech quality is generated based on unit selection (Hunt and Black, 1996; Clark et al., 2007). The basic concept of this method consists of concatenating speech segments without modification. It uses a large database, including units in different phonetic and prosodic contexts. However, large speech database requires a huge memory storage. Furthermore, to have different voice styles and emotions, another speech database is required for such style or emotion which increases the required storage capacity (Zen et al., 2009). These issues make corpus-based synthesis systems not suitable for devices with limited resources.

In direct contrast to the concatenation of pre-recorded speech units approach, Statistical Parametric speech Synthesis (SPS) approach consists of converting a set of parametric representations to speech waveform. During the recent years, SPS approach has been growing fast in popularity and the generated speech quality has been found to be as intelligible as human voice (Zen et al., 2009, 2013; Tokuda et al., 2013; Ekpenyong et al., 2014). In such SPS system, pre-recorded speech database is replaced by a set of generative models (e.g. neural networks, hidden Markov models). These techniques are used to model the acoustic parameters (spectral and excitation parameters) extracted from a speech database. Subsequently, the target speech waveform is reproduced from the appropriate speech parameters through a source-filter model. The main advantages of the SPS approach over the concatenative approach are the small memory footprint and the flexibility of voice characteristics’ modification (e.g. style, emotions) (Nose et al., 2005, 2007; Barra-Chicote et al., 2010).

Statistical parametric synthesis systems are composed of two parts: training part (generative model is used to create models that map the linguistic features into acoustic parameters) and synthesis part (reconstruct the speech waveform from the predicted parametric representations using a vocoder, e.g. MLSA-based vocoder: Mel Log Spectrum Approximation (Imai et al., 1983)). Since the last decade, Hidden Markov Models (HMMs) have been widely used in speech synthesis for many languages (Qian et al., 2006; Abdel-Hamid et al., 2006; Fares et al., 2008; Bahaadini et al., 2011; Phan et al., 2013). These models fall within the category of shallow architectures which are based on a single hidden layer of non linear transformation. In the last few years, deep learning has emerged as a new area of machine learning research (Deng and Yu, 2014). Deep learning techniques are impacting a wide range of signal and image processing applications. For instance, deep learning using neural network achieved high performance in many tasks, including speech processing (speech recognition and synthesis) (Martin et al., 2013) and computer vision (Ciresan et al., 2010). Opposed to shallow techniques, deep architectures are based on many layers of non-linear transformations. With the fast development of hardware and software, it became possible to use neural networks with

Table 1
Arabic consonants.

		Bilabial	Labio-dental	Inter-dental	Alveo-dental	Palatal	Velar	Uvular	Pharyngeal	Glottal
Stops	Voiced	ب			د,ض	ج				
	Unvoiced				ت,ط		ك	ق		أ
Fricatives	Voiced			ذ,ظ	ز			غ	ع	
	Unvoiced		ف	ث	ص,س	ش	خ		ح	هـ
Nasals	Voiced	م			ن					
Trill	Voiced				ر					
Lateral	Voiced				ل					
Semi Vowels	Voiced	و				ي				

complex architecture (multiple hidden layers) and to train the network with massive data (Ciresan et al., 2010; Deng and Yu, 2014).

The aim of this work is to develop and evaluate a TTS system based on statistical parametric approach. This system is dedicated to Arabic language and takes into account the specificities of this particular language. An automatic restoration of Arabic diacritics system is proposed to solve the problem of missing of diacritic marks. The speech database used in this work is composed of non-uniform unit segments. This choice aims at improving the speech quality. In (Zen et al., 2013), the authors mention three factors that impact directly the speech quality, which are: vocoding, accuracy of acoustic models, and over-smoothing. In this paper, we address the accuracy of diacritization and acoustic models. We propose to use deep neural networks. It was shown that this networks outperform the traditional HMM-based synthesis system (Zen et al., 2013). Moreover, we develop the diacritization system based on deep learning techniques which were not applied to the task of diacritics recognition. Along with the use of deep architecture, different methods are proposed in order to increase the accuracy of diacritization and acoustic models.

This paper is organized as follows. Arabic language's characteristics are discussed in Section 2. Section 3 gives an overview of the text-to-speech framework. The proposed diacritization system is detailed in Section 4. Section 5 describes the deep neural network based speech synthesis. Evaluation results of the TTS system are presented in Section 6. Conclusion and future work are presented in Section 7.

2. Arabic language

2.1. Arabic phonetic system

The phonetic system of Modern Standard Arabic has 34 phonemes: 26 consonants, 3 short vowels (Fatha ـَ , Dhamma ـُ , Kasra ـِ), 3 long vowels (ا , و , ي) and 2 semivowels (و , ي) depending on the context which they appear in the word. Arabic writing system consists of 36 letters representing the consonants. Each letter represents a single consonant with some exceptions (some consonants are written in different forms depending on their position in the word, e.g. ت , ة). Arabic diacritical signs are the vowelization marks (Sukun, Fatha, Dhamma, Kasra), the gemination mark (shadda) and the suffixes, known as tanween signs (an ـً , on ـٌ , in ـٍ). Sukun mark is usually not written because a consonant without a vowel is considered with sukun. The International Phonetic Alphabet (IPA) classifies the Arabic consonant according to the place and manner of articulation. Table 1 shows Arabic consonants based on the IPA classification.

2.2. Arabic syllables

Arabic syllables are a number of six which are: CV, CVV, CVC, CVCC, CVVC and CVVCC, where C stands for consonant and V stands for vowel. CV is the frequent syllable, whereas CVVCC is very rare. There is two kinds of syllables: open syllables which finish with vowel (e.g. CV) and closed syllables that finish with consonant (e.g. CVC).

Table 2
Statistics of occurrence of the Arabic word "كتب" with different diacritics based on Google search engine.

Arabic word	Frequency	Percentage
كتب	94 600 000	91.71%
كَتَبَ	4 640 000	4.49%
كُتُبٌ	3 930 000	3.8%
Total	103 170 000	100%

Every syllable begins with a consonant followed by a vowel (short or long vowel). The vowel is the nucleus of the syllable. Arabic words contain one or more syllables. The syllables in a word are pronounced with different stress levels. The Arabic studies in prosody defines three lexical stress levels for each syllable in a word: primary stress (PS), secondary stress (SS) and weak or unstressed level (US). The identification of the syllable stress levels in an Arabic word depends on a set of rules which are as follows (Chouireb and Guerti, 2008; Zaki et al., 2010):

- If the word is composed of only CV syllables, the first syllable gets the primary stress and the remaining syllables receive unstressed level. For instance, ذَهَبَ in English "he went" has these lexical stresses: ذَ (CV → PS) هَ (CV → US) بَ (CV → US).
- If the word contains only one long syllable, the long syllable receives the primary stress and the rest of syllables are unstressed. Example, نَامَ ("sleep"): نَا (CVV → PS) مَ (CV → US).
- If the word contains two or more long syllables, the long syllable nearest to the end of the word receives the primary stress, the long syllable nearest to the beginning gets the secondary stress and the remaining syllables are unstressed. Example, the word كِتَابَاتُهُمْ ("their writings"): كِ (CV → US) تَا (CVV → SS) بَا (CVV → PS) تُ (CV → US) هُمْ (CVC → US).

In Arabic, the last syllable in a word never gets a primary or secondary stress, it is always unstressed. The monosyllabic prepositions receive a secondary stress instead of a primary stress.

2.3. Arabic diacritizer

Arabic language is a diacritized language whose alphabet contains only consonants and the diacritic marks are added to specify the pronunciation of the text. Unfortunately, modern Arabic text, books, journals, and internet documents are written without mentioning these marks (Attia, 2005; Badrashiny, 2009; Hamad and Hussain, 2011). Table 2 gives the statistics of the occurrence of the Arabic word "كتب" with different diacritics. Based on similar results of other Arabic words, undiacritized Arabic words existing on the internet nowadays are more than 90%.

The readers restore the missing diacritic marks based on their knowledge of the language and the context while reading an undiacritized text. In case of a severe ambiguity, the writer puts the diacritics to overcome any problem. However, automatic processing of Arabic text is often hampered by the lack of diacritic signs. For instance, an Arabic TTS would not generate speech from undiacritized text because there are different pronunciations of the same undiacritized word. For instance, the word علم , when diacritized, could be: عِلْم (science), عَلَم (flag) and عَلَّمَ (he taught). Moreover, the process of syllabification of the input text cannot be performed without diacritic marks. Many Arabic diacritization systems were developed and different methods were proposed. There are few known commercial systems

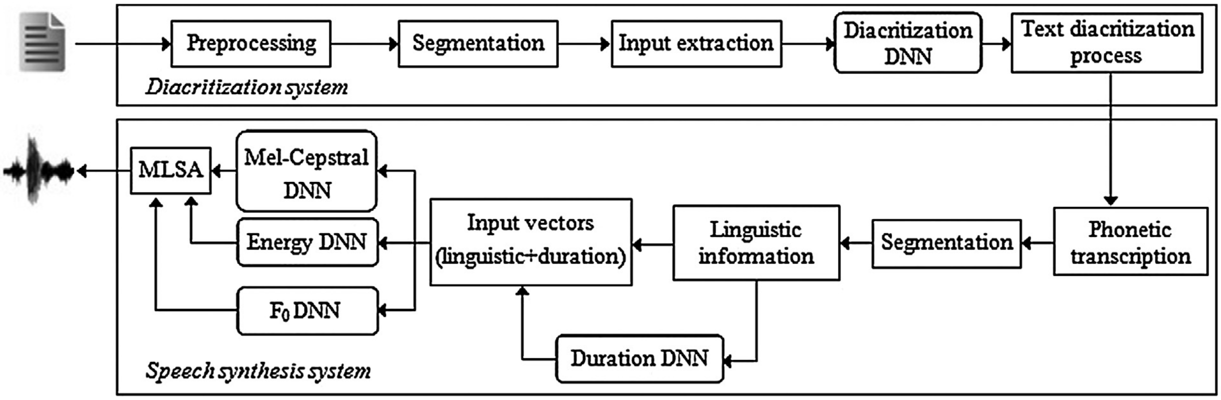

Fig. 1. Text-to-speech synthesis system for Arabic language based on neural networks.

(RDI,[1] Sakher,[2] Cimos[3]) whose source code is not available. They are generally used as black boxes in context specific applications. Other non-commercialized systems were developed by researchers using personalized methods. There are three major family of techniques: *dictionary method*, *rule-based method* and *machine learning method*.

The first technique is based on the standard Arabic dictionaries. It consists of building a large database of Arabic vocabulary. For each undiacritized word, the system searches the words with the same spelling. Then, a disambiguation phase is performed in order to choose the right diacritized word. Sakher system is based on this method. Although this technique has the advantage of being accurate, it is costly in terms of database collection (difficult to collect all Arabic vocabulary), search process (high access time in the search process) and memory storage capacity (Badrashiny, 2009). Moreover, it has a coverage problem: the largest dictionary may not have some morphologically possible vocabularies (e.g. non used words in some Arabic countries can be used in other countries).

Unlike dictionary method, rule-based diacritization systems have the advantage of not being attached to a fixed vocabulary. It applies a complex combination of morphological, syntactic and semantic rules. The morphological analyzer decomposes the undiacritized word into its morphological entities using known patterns or templates. These morphological entities are represented by the quadruple Q = (t: p, r, f, s), where: t is the class of the word (regular derivative, irregular derivative, fixed, or arabized), p is prefix, r is root, f is form and s signifies the suffix. Syntactic rules are then used to determine the case-ending diacritics. Semantic rules help to resolve ambiguous cases. RDI system is developed using this technique. The main drawbacks of this method are the high complexity and the long-timed processing of a given Arabic sentence (Badrashiny, 2009).

The third method consists of using machine learning techniques (e.g. HMM, neural networks) to generalize knowledge extracted from a diacritized text to new situations. A generative model is used to model the diacritic marks extracted from a full diacritzed text. Afterward, the system predicts the diacritic marks of the undiacritized text based on the created model. This approach has many advantages over the above-mentioned ones, such as small memory footprint and short time for processing a sentence. HMM is the most common machine learning technique (Elshafei et al., 2006; Khorsheed, 2012).

Our aim is to build an automatic diacritization system with small memory footprint, low computational cost and high efficiency. Thus, we propose a diacritization system based on DNN.

3. Arabic text-to-speech system

The overview of the Deep Neural Networks (DNN) based system for Arabic TTS is shown in Fig. 1. Our complete TTS system is composed of two sub-systems. The first one is a diacritization system which is designed to restore the

[1] Research & Development International (RDI): http://www.rdi-eg.com.

[2] Sakhr: http://www.sakhr.com.

[3] Cimos: http://www.cimos.com.

missing diacritic marks of the Arabic text. The second one is a speech synthesis system that transforms the resulting text into speech waveform.

The diacritization system is composed of three parts: text-to-linguistic engine (including preprocessing phase, segmentation of the text and input feature extraction module), DNN model to recognize the diacritic mark of each input vector and the diacritization module. The preprocessing phase is very important. In fact, the text is analyzed in order to remove all signs of ambiguity that degrades the performance of the diacritizer system. Therefore the performance of the synthesizer is reduced. The resulting text, a full diacritized Arabic text, which is, then, the input of the second system to generate speech waveform.

The speech synthesis system consists of: a phonetic transcription engine, linguistic features extraction module, duration DNN used to predict the duration and the temporal trajectory of each acoustic unit, another three DNNs used to transform the input features into prosodic and spectral parameter vectors (fundamental frequency (F_0), energy and Mel-cepstral coefficients) and finally MLSA vocoder (Imai et al., 1983).

4. Diacritization system

The diacritization system is developed in order to provide the missing diacritic signs. It is based on DNN. Our algorithm generates diacritized text with determining the final diacritic mark of the word. First, a set of preprocessing steps needs to be fulfilled:

- Preprocessing on non lexical elements: transform any non orthographic sign by its meaning in Arabic words. The text analyzer examines all the text and expands digital and numerals into full Arabic words (e.g. numerals, dates, abbreviations, symbols, etc.).
- Removing any character other than Arabic letters and punctuation.

The resulting text obtained after this process is entirely orthographic. Next, the text is segmented to sentences based on the punctuation marks and then each sentence is segmented to words using the space character. Finally, the letters of each word are transformed into numerical feature vectors through the linguistic feature extraction module.

4.1. Experimental conditions

For training and testing purpose, we used a fully diacritized Arabic corpus. The corpus was collected from different Arabic books and articles covering various subjects. Besides, the text corpus was manually checked by Arabic language specialist to correct partially diacritized words. The total number of words in our corpus is 25k words. The corpus was divided in three sets as follows: training set (80%), development set (10%) and test set (10%).

To diacritize the Arabic text, our diacritization system is expected to recognize all the Arabic diacritical signs: the vowelization marks (Sukun, Fatha ـَ, Dhamma ـُ, Kasra ـِ), the gemination mark (shadda) and the tanween signs (an ـً, on ـٌ, in ـٍ). Two neural networks were used to model them. The first neural network was used to predict seven diacritic marks: the four vowelization marks and the three tanween signs. Six parameters were selected to perform this task with high performance. The input parameters could be grouped into two categories: specific parameters related to the current letter (e.g. current letter orthography, letter position in the word) and contextual parameters (adjacent letters orthography). Table 3 presents the input parameters used to predict the diacritic marks.

The number n of adjacent letters was determined by testing different levels of adjacent in order to select the optimal number. Multiple neural networks were trained and tested for each adjacent level: multiple hidden layers with different hidden units per layer. The neural networks were trained for 200 iterations. Diacritization evaluation of our experiments is reported in terms of Diacritization Error Rate (DER). To recognize the diacritic mark generated by the DNN, we used the Arg max function which stands for the argument of the maximum. The predicted class $\hat{\imath}$ would be:

$$\hat{\imath} = \underset{i}{\operatorname{argmax}} \quad a_i \tag{1}$$

Fig. 2 gives the DER values relative to each neural network for each adjacent level (3-adjacent, 4-adjacent and 5-adjacent).

Table 3
Neural network input features

Input information	Input values	Input layer nodes
Current Letter (CLt)	ا,ب,ت,ة ... ي,و,ه,	36 nodes
n-adjacent Left and Right Letters (n-LfL, n-RtL)		2*n*36 nodes
Current Letter position in Word (CLW)	[0..1]	1 node
Current Word position in Sentence (CWS)		1 node
Sign shadda (SS)	Yes/no	1 node

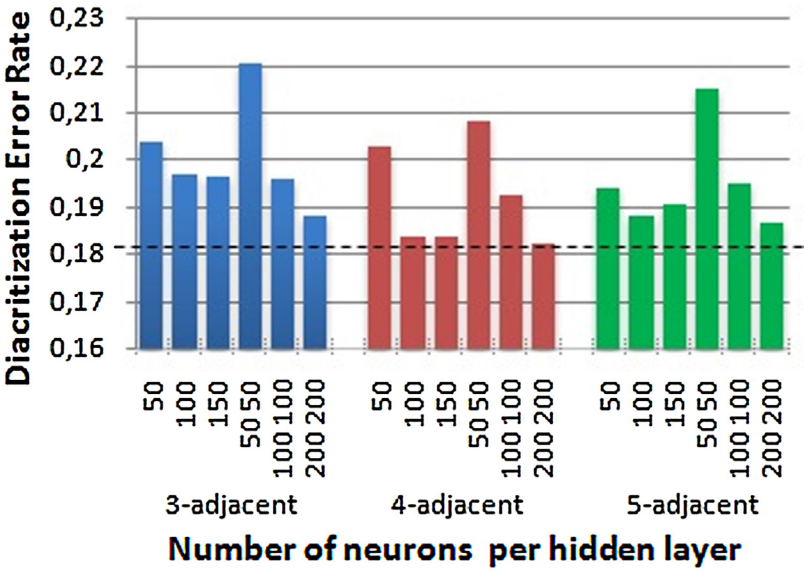

Fig. 2. DER values and the corresponding neural network architectures.

It can be seen that almost all the experimental results of the diacritization system based on four adjacent letters was better than those obtained by three and five adjacent letters. Thus, training data based on four adjacent letters features was used for further experiments.

The second neural network was used to recognize the gemination mark *shadda*. This network was trained using the same input parameters in Table 3 with the exception of the parameter *sign shadda* which was used as output parameter rather than input parameter. The process of selection of the number *n* was performed and 4-adjacent letters gave the best results.

Table 4 gives an example of coding the input and output parameters used in the diacritization system of the letter "ك" of the word "شُكرًا" ("thanks") with one left and right adjacent.

Table 4
Encoding of the letter "ك" of the word "شُكرًا".

Input						Output
1-LfL	CLt	1-RtL	CLW	CWS	SS	Diacritic
ش 0..010..0	ك 0..100.0	ر 00..10.0	0.334	0	0	- 1-1-1-1-1-1-1

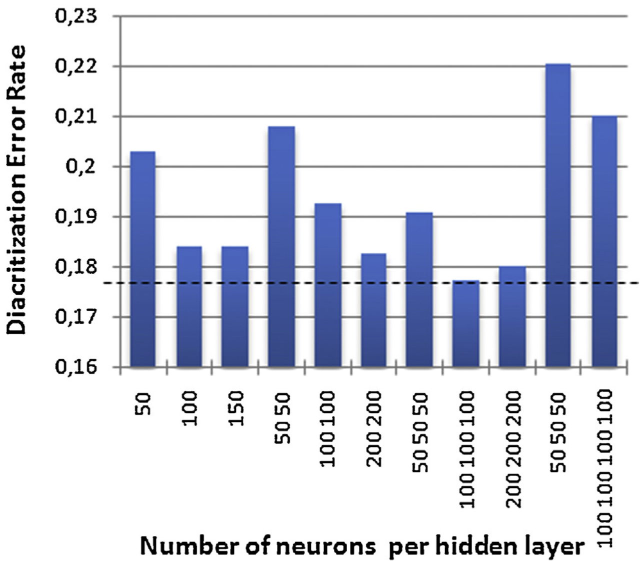

Fig. 3. DER values and the corresponding neural network architectures.

4.2. Experimental results

4.2.1. All diacritic class with one neural network

Two DNNs were used to predict the different Arabic diacritic marks. We trained many neural networks by varying the number of hidden layers and the numbers of hidden units (50, 70, 100, 200 units per layer). The neural networks were trained for 200 iterations. The architecture of the network used to recognize the shadda mark consists of three layers: 326 input nodes, 150 neurons in the hidden layer and 1 output node. Both hidden and output layers used tangent sigmoid transfer function. The shadda sign recognition is based on a threshold β: if neural network output is above β then the current letter is geminated, not otherwise. By varying the threshold values, the best β value was found to be 0 where FRR = 0.01 (False Rejects Rate) and FAR = 0.009 (False Alarms Rate) using the development set.

The second neural network was used to model seven diacritic marks. Mostly but not always the number of hidden units was the same in all hidden layers. Fig. 3 presents the DER values and the corresponding neural network architectures. Note that the DER values correspond to the database using four adjacent letters and the original shadda information.

The architecture of the multilayer network consists of four layers: 327 input nodes, 100 neurons in all hidden layers and 7 output nodes. Both hidden and output layers use tangent sigmoid transfer function. The DER values are equal to 17.7% and 18.2% without and with using the DNN shadda recognition model respectively.

4.2.2. Multi neural networks for sub-class

Arabic language is a rule based language in which the diacritization of the Arabic text obeys known rules. We chose three rules in order to adopt our system based on these rules, which are:

- The first letter of all Arabic words, derivative or non-derivative words, never begins with sukun. Just one of the short vowels (ـَ , ـُ , ـِ) can be used.
- The letters in the middle of a word are followed by the vowelization marks: short vowels along with sukun sign. Thus, four diacritics can be used.
- The final letter is followed by all the Arabic diacritic marks.

Taking into account these rules, we proposed to build three neural networks rather than using one neural network to predict the vowelization and tanween marks. Based on the position of the current letter in the word, we used a network to predict the short vowels (Fatha, Dhamma, Kasra) for letters in the beginning position, a network to predict the four vowelization marks for letters in the middle and another network to predict all the diacritic marks for final letters. The architecture of the proposed method is shown in Fig. 4.

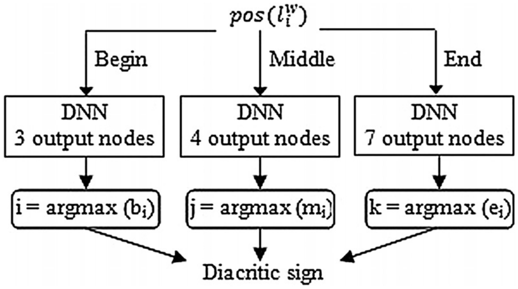

Fig. 4. Multi neural network based diacritization system.

Table 5
Text conversion using our diacritization systems.

Original text: إِيَّاكَ وَالوُقُوعَ فِي الشُّبُهَاتِ وَالوُلُوعَ بِالشَّهَوَاتِ فَإِنَّهُمَا يَقتَدَانِكَ إِلَى الوُقُوعِ فِي الحَرَامِ وَرُكُوبَ كَثِيرٍ مِنَ الآثَامِ
Diacritized text using M1: إِيَّاكُ وَالوقُوعِ فِي الشَّبَهَاتَ وَالولُوعَ بَالشَّهُوَاتِ فَإِنَّهِمَا يَقتَدَانِك إِلَى الوَقُوعَ فِي الحَرَامِ وَرَكُوبَ كَثِيرٌ مِن الآثَامُ
Diacritized text using M2: إِيَّاكَ وَالوَقُوعَ فِي الشَّبِهَاتِ وَالوُلُوعَ بِالشَّهَوَاتِ فَإِنَّهُمَا يَقتَدَانِكَ إِلَى الوَقُوعِ فِي الحَرَامِ وَرَكُوبَ كَثِيرَ مِن الآثَامِ

$pos(l_i^w)$ denotes the position of the letter l_i within the word w. This method has improved the DER value compared with the previous method by 1.9% using the original diacritic shadda and by 1.3% using DNN shadda model. The diacritic recognition rate is equal to 83.1%.

In Table 5, we give an application of our diacritization system. We present the original diacritized text and the diacritized text obtained using our first method M1 and using the modified approach M2. Obviously, the second system provides a more accurate text than the one provided by the first system.

5. Speech synthesis system

Our statistical parametric synthesis system consists of converting a set of linguistic features into a set of acoustic parameters. DNNs are used to model the acoustic parameters. An Arabic speech data of pre-recorded speech from a female speaker was used to train the DNNs. It consisted of about 1030 sentences. These sentences were composed of 843 declarative sentences, 117 interrogative sentences and 70 exclamatory sentences. In addition, they contained 3732 words with 3154 distinct words. We used 927 sentences for training while the remaining sentences were used for the evaluation. The speech data was down-sampled from 44.1 kHz to 16 kHz sampling and for each 25ms frame length and 10ms frame shift, Mel-cepstral coefficients (39 coefficients including the zero-th coefficient) and F_0 were computed

Table 6
Different type of units for the character “ب ”.

Character “ب ”						
C	CV	CV	CV	CVV	CVV	CVV
ب	بَ	بِ	بُ	بَا	بِي	بُو

through Speech Signal Processing Toolkit.[4] The unvoiced frames were interpolated and modeled as voiced frames. Furthermore, the duration was also extracted from each segment. It was used to ensure the naturalness of speech.

5.1. Data segmentation

Arabic speech synthesis systems face several problems (e.g. speech quality, articulatory effect, discontinuity effect). Different methods have been proposed to solve these problems, such as: Maximum Likelihood Parameter Generation (MLPG) (Raghavendra et al., 2010) to obtain smoother trajectories and to reduce the discontinuity effect, the interpolation features (Chouireb and Guerti, 2008; Raghavendra et al., 2010), the use of large and different unit types method which is mainly used in concatenative synthesis systems (Elshafei et al., 2002; Al-Said and Abdallah, 2009; Hamad and Hussain, 2011). In (Hamad and Hussain, 2011), the authors developed an Arabic text-to-speech that uses allophone and diphone concatenation method. The variety in speech segments allows the generation of more natural speech. However, this method requires a large number of units, to build a basic corpus.

In this work, we used two methods: the interpolation features as input of the neural network and varied length units database. The challenge was to choose unit types that require an acceptable number of units (to build a basic units corpus) and minimize the articulatory effects. We proposed as units: phonemes (Arabic consonants), two-phonemes (two successive phonemes) and three-phonemes (three successive phonemes). Our method consists of combining Arabic vowels with consonants in order to minimize the articulatory effects. Moreover, it requires an acceptable number of units. Each type of unit must have only one consonant and with or without vowels depending on the context. The segmentation rules are the following:

1 If a consonant is followed by another consonant, the first consonant represents a unit (C).
2 If a consonant and a vowel are followed by another consonant, the first consonant and the vowel represent a unit (CV).
3 If a consonant, a short vowel and a long vowel are followed by another consonant, the first consonant and the two successive vowels represent a unit (CVV / CV:).

A base units corpus contains 196 units: 28 C units + 28*3 CV units + 28*3 CVV units where 28 are the number of Arabic consonants, 3 short vowels and 3 long vowels. We show an example of the character “ب ” in different units cases in Table 6.

Moreover, our method reduces the number of segments composing a word. Fig. 5 presents a segmentation of the word “شُكرًا” (“thanks”) with phonemes and our different units. It can be seen that the word “شُكرًا” is composed of 7 phonemes while it needs 4 units with our segmentation system.

The segmentation of the speech database into units gave 12875 segments with: 7725 CV units, 3399 C units and 1751 CVV units. The correction of the segmentation was based on listening of speech signal.

5.2. Neural networks input features and evaluation

The performance of the neural networks to map from linguistic to acoustic parameters depends in large part on the input information. The input features for the DNN based synthesis system included the context of the current, the

[4] SPTK: ”Speech Signal Processing Toolkit”, (http://sp-tk.sourceforge.net/), Version 3.6, 2012.

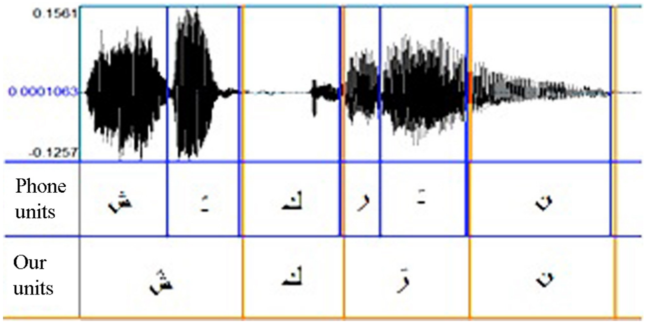

Fig. 5. Our segmentation system and phoneme segmentation system.

previous and the next unit (unit features, articulatory features, syllables features) and the positions (e.g. unit position, syllable position). Along with these contextual features, 55 numerical features for coding the relative position of the current frame within the unit and one feature for duration of the current unit were used. The input used to train the neural networks included 191 values with 135 binary features and 56 numerical features. The input features used in our work are detailed in Table 7. To model the parameter duration, the input features presented in Table 7 had been used with the exception of the parameters; *temporal variations* and *unit duration* which were removed. The temporal variation features was used to indicate the position of the current frame within the current unit and to ensure a smooth transition between neighboring frame vectors, especially between neighboring vectors on segment boundaries. Fifty-five time index neurons were used to represent the temporal variations. The value of time index i during frame j is computed using the Eq. (2) (we chose $\beta = 0.2$ after experimental tests) (Chouireb and Guerti, 2008) and time index i reaches its maximum value (= 1) when frame $j = i$.

$$Oi = \exp(-\beta(i - j)^2) \tag{2}$$

To evaluate objectively the performance of DNN-based synthesis system, Mel-Cepstral Distortion (MCD) (Zaki et al., 2010) (Eq. (3)) and Root Mean Squared Error (RMSE) (Eq. (4)) had been used to predict the accuracy of acoustic models.

$$MCD = \frac{10}{\ln 10} * \sqrt{2 * \sum_{i=1}^{39} (mcep_i^t - mcep_i^e)^2} \tag{3}$$

$$RMSE = \sqrt{\sum_{i=1}^{T} (x_i^t - y_i^e)^2 / T} \tag{4}$$

where $mcep_i^t$ and $mcep_i^e$ are the target and the estimated Mel-cepstral coefficients while x_i^t and y_i^e are the target and the estimated values of F_0, energy and duration.

5.3. Experimental results

The purpose of this part is to build neural network models which generate automatically Mel-cepstral coefficients and prosodic parameters (energy, F_0 and duration). The DNNs should learn the example of the training data and without losing the ability to generate new situations. Both input and output features in the training data were not normalized. Different methods were proposed to model the acoustic parameters based on acoustic and linguistic characteristics and based on deep learning techniques (stacked generalization techniques). So, we used the conventional methods which consist of modeling the parameters using either single network for all parameters or neural network per parameter. Along with these, we proposed DNN-based systems using units length and using vowelization marks. Finally, we proposed a deep learning architecture based on the stacked generalization technique by stacking two neural networks.

Table 7
Input features for neural networks: a-overall features, b-unit articulation features, c-unit features, d-syllable features, e-other features.

Features	Nodes
(a)	
Current unit features	9 (bin)
Previous unit features	9 (bin)
Next unit features	9 (bin)
Current unit articulatory features	25 (bin)
Previous unit articulatory features	25 (bin)
Next unit articulatory features	25 (bin)
Current syllable features	6 (bin)
Previous syllable features	6 (bin)
Next syllable features	6 (bin)
Current unit position in the syllable	3 (bin)
Current unit position in the word	3 (bin)
Current syllable position in the word	3 (bin)
Current word position in the sentence	3 (bin)
Type of sentence	3 (bin)
Temporal variations	55 (float)
unit duration	1 (float)

Features	Features values	Nodes
(b)		
Unit type	C/CV/CVV	3 (bin)
Consonant voiced	Voiced/Unvoiced	2 (bin)
Consonant fluency	Fluency/Desisting	2 (bin)
Consonant plosiveness	Plosiveness/ Fricativeness/ Middle	3 (bin)
Consonant height	Elevation/Lowering	2 (bin)
Consonant adhesion	Adhesion/Separation	2 (bin)
Lip rounding	Yes/No	2 (bin)
Place of articulation	Bilabial/ Labiodental/ Interdental/ Alveodental/ Palatal/ Velar/ Uvular/ pharyngeal/ laryngeal	9 (bin)
(c)		
Letter	ا,ب,ت, ... ,ه,و,ي	5 (bin)
Diacritic marks	Fatha/ Dhamma/ Kasra/ Sukun	4 (bin)
(d)		
Syllable type	cv/ cvc/ cvv/ cvcc/ cvvc/ cvvcc	3 (bin)
Syllable stress level	Primary stress/ secondary stress/ unstressed	3 (bin)
(e)		
Unit position in syllable	Begin/ middle/ end	3 (bin)
Syllable position in word	Begin/ middle/ end	3 (bin)
Unit position in word	Begin/ middle/ end	3 (bin)
Word position in sentence	Begin/ middle/ end	3 (bin)
Types of sentence	Declarative/ Interrogative/ Exclamatory	3 (bin)

5.3.1. One neural network for Mel-cepstral coefficients, energy and F_0

In this experiment, we modeled the parameters: energy, 39 Mel-cepstral coefficients and F_0 with one network. The parameter duration was trained with separate network. The architecture of our network in this method is based on 3 layers: 191 input nodes, 250 hidden nodes and 41 output nodes (39 Mel-cepstral coefficients + energy + F_0). Linear function was used in the output layer while sigmoid function was applied in the hidden layer. The MCD value obtained is 7.95, RMSE value for F_0 is 0.19 and RMSE for energy is 0.85. The architecture of the duration neural network is: 126 input nodes, 10 hidden nodes (sigmoid function) and one output neuron (sigmoid function) and RMSE is equal to 0.038.

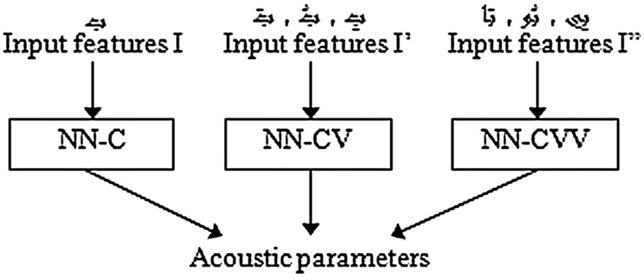

Fig. 6. Separate network for each unit type.

5.3.2. *Neural network per parameter*

We interpreted that the mapping of Mel-cepstral coefficients, energy and F_0 with one neural network gave a higher error rate for all predicted parameters. The large difference in intervals between these parameters degraded the prediction performance of the neural network. Thus, we used a separate neural network for each parameter: Mel-cepstral coefficients, energy, duration and F_0. The neural network architectures are as follows: Mel-cepstral coefficients: 191 L 20 G 100 G 39 L with MCD = 7.52, Energy: 191 L 50 S 1 L with RMSE = 0.7, F_0: 191 L 10 S 30 S 1 L with RMSE = 0.1. The numbers denote the number of nodes in the corresponding layer and the letters denote the activation function: linear (L), gaussian (G) and sigmoid (S). The duration network had the same architecture of the previous experiment. It can be seen that the use of a network for each parameter improved tremendously the accuracy of acoustic models comparing with the previous method.

5.3.3. *Separate neural network for each unit type per parameter*

As we used an inventory with varied unit types, which are: phonemes, two-phonemes and three-phonemes, we informally noted that the combination of all unit types to perform a mapping from linguistic to acoustic parameters may cause a problem. Thus, we proposed to separate unit type into three sub-classes: phonemes (C), two-phonemes (CV) and three-phonemes (CVV), and for each sub-class training data we used a neural network. Fig. 6 shows the architecture of the proposed method.

In this experiment, 41 time index neurons, 45 time index neurons and 55 neurons were used to represent the temporal for unit types: phonemes, two-phonemes and three-phonemes respectively. Thus, the input vector used to train the networks ANN-C, ANN-CV and ANN-CVV contained 177, 181 and 191 input features respectively. Note that the input vector of the parameter duration left unchanged for all types of network.

Briefly, the architectures of the neural networks used in this method are as follows:

- Mel-cepstral coefficients: ANN-C: 177 L 20 G 100 G 39 L, ANN-CV: 181 L 20 G 120 G 39 L, ANN-CVV: 191 L 30 G 150 G 39 L. The MCD value decreased from 7.52 to 7.21.
- Energy: ANN-C: 177 L 30 S 1 L, ANN-CV: 181 L 45 S 1 L and ANN-CVV: 191 L 45 S 1 L. RMSE value is 0.59.
- F_0: ANN-C: 177 L 10 G 35 G 1 L, ANN-CV: 181 L 12 G 40 G 1 L and ANN-CVV 191 L 20 G 40 G 1 L. RMSE value is 0.04.
- Duration: 126 L 10 S 1 S for all networks, ANN-C, ANN-CV and ANN-CVV. RMSE value is found to be 0.03.

These results show that multiple neural networks method based on unit type gave better results.

5.3.4. *Separate network for each diacritic mark per parameter*

In this experiment, practically the same idea was applied as the previous method. We divided the data in sub-groups, but instead of using the type of units, we proposed to use the vowelization marks. Hence, we divided the data into four sub-classes: Sukun, Fatha, Dhamma and Kasra, and four neural networks were used. The proposed method is presented in Fig. 7. The models of the network (sukun) were the same obtained in our previous experiment, since the data did not change. The input vectors which had been used to train the other networks, NN (Fatha), NN (Dhamma) and NN (Kasra), contained 191 values for parameters Mel-cepstral, energy and F_0.

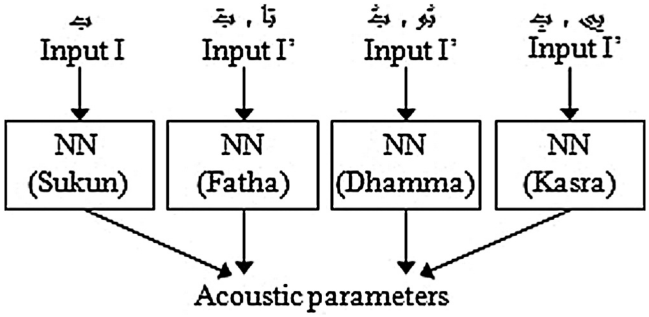

Fig. 7. Separate network for each diacritic mark per parameter.

The experimental results were slightly higher than the previous method: MCD value was found to be 7.25, the RMSE of energy models was equal to 0.61, F_0 RMSE value was equal to 0.06 and the RMSE of parameter duration was found to be 0.032. We can observe that the classification of data based on vowelization marks gave better results than conventional methods (neural network per parameter) but higher than the classification based on unit types.

5.3.5. *Recursive neural networks*

To improve the accuracy of the acoustic models, we proposed a simple stacked generalization technique using neural networks. Our idea consisted of building deep architecture using neural networks and stacking technique. The overall architecture of our approach is shown in Fig. 8.

The outputs of each layer (neural network outputs) are fed to the immediate layer as input along with the input features. In this paper, we present a simple architecture based on two stacked layers (called Recursive Neural Networks RNN). The outputs of the first network are used along with other input features to train the second neural network. The input features of the second network are:

- *Neural network outputs*: 24 Mel-cepstral, energy, F_0 and duration.
- *Linguistic features*: previous, current and next unit features. These features are added to improve more the performance.

The first neural network architectures were those obtained by using neural network per parameter method. The second neural network architecture are as follows: Mel-cepstral: 66 L 150 S 39 L with MCD = 7.32, Energy: 28 L 10 S 1 L, with RMSE = 0.66, F_0: 28 L 13 S 1 L with RMSE = 0.07 and Duration: 28 L 10 S 1 L with RMSE = 0.035. Our recursive neural networks method improved the accuracy of acoustic models over the conventional methods, but didn't outperform unit types method.

5.4. *Results*

Based on the experimental results of all the previous proposed methods, DNN-based separate network for each unit type speech synthesis outperformed conventional methods, separate network for each diacritic mark and stacking technique. The trajectories of 1-th Mel-cepstral coefficients and energy of original and predicted speech by the system based on the best method of the Arabic sentence “بِخَيْرٍ، شُكْرًا” (Fine, thank you) are shown in Fig. 9 and Fig. 10 respectively.

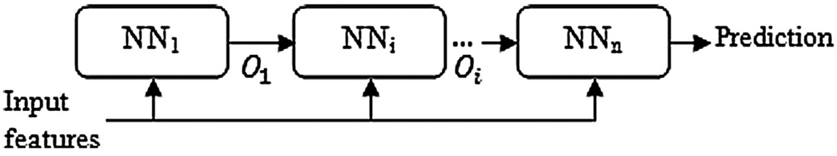

Fig. 8. Recursive neural network architecture.

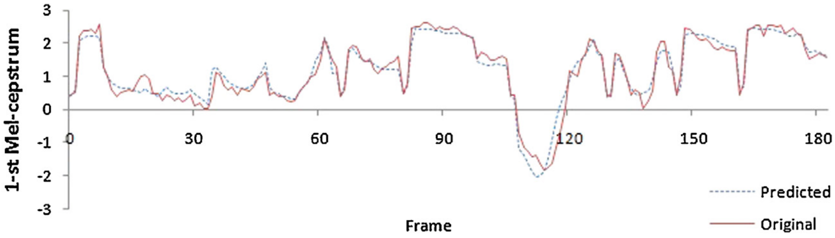

Fig. 9. Trajectories of 1-st Mel-cepstral coefficients of original speech and those predicted by M3.

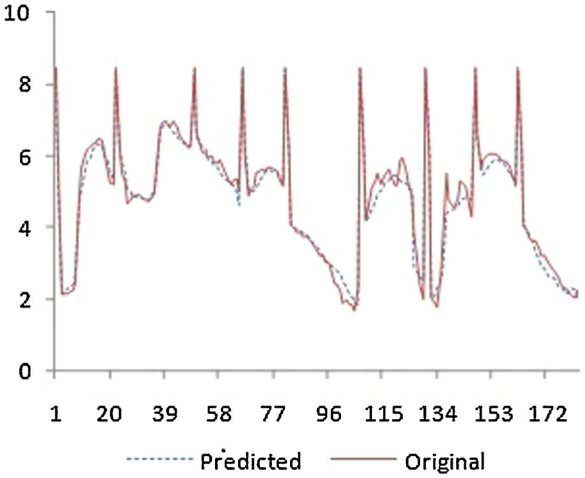

Fig. 10. The predicted and natural energy trajectories.

It can be seen from these figures that the predicted values follow almost the same trajectory as the original values. However, small details are smoothed. Fig. 11 shows the waveform, spectrogram and F_0 contour of the Arabic sentence “بِخَيرٍ، شُكرًا”: part(a) shows the original speech waveform, part(b) shows the synthesized speech using the system based on unit type method. The synthesized speech waveform was generated based on 39 Mel-cepstral coefficients, energy and F_0 through MLSA vocoder and using the predicted duration values. Comparison between the waveform of the original and synthesized speech shows smooth transitions between segments and especially at segment boundaries.

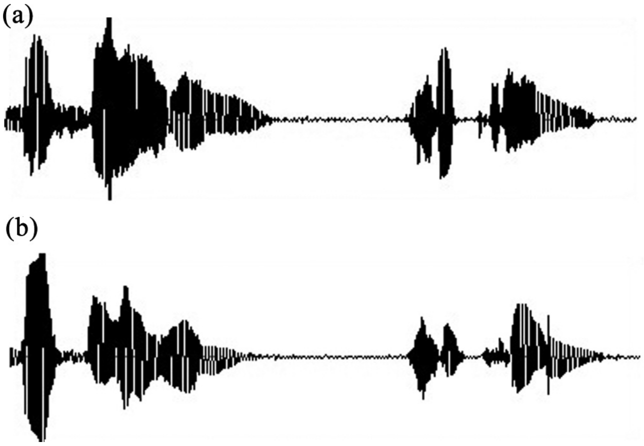

Fig. 11. (a) Original speech waveform. (b) synthesized speech system.

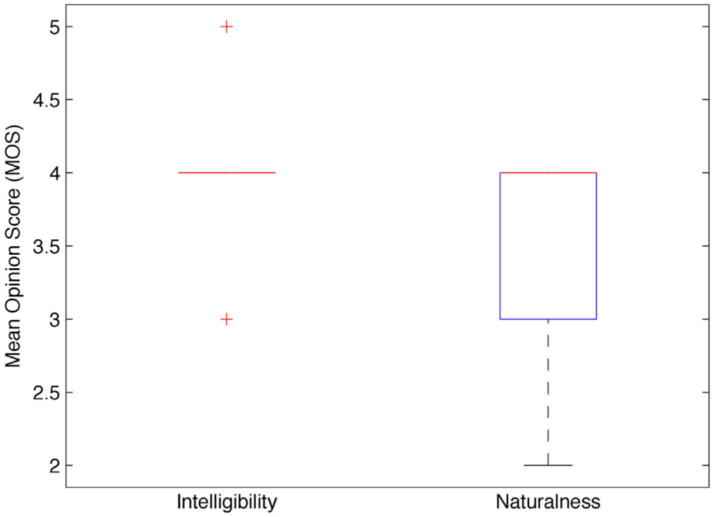

Fig. 12. Intelligibility and naturalness evaluation.

6. Evaluation

For subjective evaluation, a listening test was conducted in order to evaluate the performance of our system. The subjects who participated in this experiment were twenty Arabic native speakers to evaluate ten sentences selected from the test set. They had no hearing problem and had a good knowledge of Arabic. The listeners were asked to rate the selected sentences independently for intelligibility and naturalness. All test sentences were played consecutively for each listener. Then, for each evaluation test, each listener gave his judgment based on the Mean Opinion Score (MOS) rating scale (score between 1 (Bad) to 5 (Excellent)). Fig. 12 presents the detailed results for both intelligibility and naturalness.

For the intelligibility scores, the values were from 3 to 5 and the greater part of subjects gave a rank of 4 for our synthesized speech. The average score over all listeners is equal to 3.9. For the naturalness evaluation, the greater part of subjects gave a rank of 3 and 4, while two subjects selected a score equal to 2. The naturalness average score is 3.65.

Another listening test was performed to evaluate how much the speech is easy to be understood by the users. In this test, each sentence was played and the listeners transcribed what they heard. If the listener didn't recognize the sentence, the sound was played again. The sentences were played up to three times. Fig. 13 plots the results of recognition of each sentence.

The average of recognition rate over all listeners and sentences was found to be 93.5%. Thus, it can be seen that our system generates good speech quality that can be easily understood.

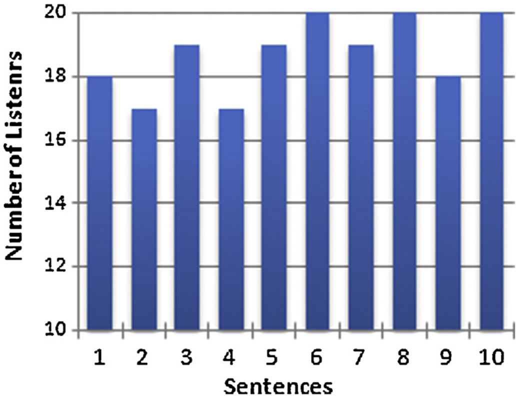

Fig. 13. Number of listeners recognizing the corresponding sentences.

7. Conclusion

This paper described a speech synthesis system for Arabic language based on statistical parametric approach. An Arabic diacritization system was proposed in order to solve the problem of missing of Arabic diacritic marks. Both systems (diacritization and synthesis systems) were based on deep learning using multilayer neural networks. To improve the speech quality, different methods have been proposed, including the use of varied unit length speech database, mainly implemented with concatenative based systems. In this work, the speech database was composed of non-uniform units segment to improve the speech quality.

In this paper, we addressed the accuracy of diacritization and acoustic models. Along with the use of DNNs, different methods were proposed in order to increase the accuracy of the diacritization and acoustic models. Furthermore, we proposed a diacritization system based on the position of the current letter. This new method outperforms the diacritization system based on a single neural network for all diacritic marks. In the synthesis system, we proposed different methods, such as neural network per unit type based system, diacritic marks based system and deep architecture using stacked generalization technique for speech synthesis. Neural network per unit type based synthesis system gave the best results. The experimental results showed that our system for Arabic TTS generates intelligible and natural speech which can be easily understood. Future work includes the improvement of the F_0 modeling schema and the use of many layers in our recursive neural networks method. Furthermore, we plan to change the current vocoder by STRAIGHT vocoder.

References

Abdel-Hamid, O., Abdou, S., Rashwan, M., 2006. Improving Arabic HMM based speech synthesis quality. In: International Conference on Spoken Language Processing, pp. 1332–1335.

Al-Said, G., Abdallah, M., 2009. An Arabic text-to-speech system based on artificial neural networks. J. Comput. Sci. 5, 207–213.

Attia, M., 2005. Theory and Implementation of a Large-Scale Arabic Phonetic Transcriptor, and Applications. Ph.D. thesis. Department of Electronics and Electrical Communications, Faculty of Engineering, Cairo, Egypt.

Badrashiny, M., 2009. Automatic diacritizer for Arabic texts. Ph.D. thesis, University of Cairo, Cairo, Egypt.

Bahaadini, S., Sameti, H., Khorram, S., 2011. Implementation and evaluation of statistical parametric speech synthesis methods for the Persian language. In: International Workshop on Machine Learning for Signal Processing, pp. 1–6.

Barra-Chicote, R., Yamagishi, J., King, S., Montero, J., Macias-Guarasa, J., 2010. Analysis of statistical parametric and unit selection speech synthesis systems applied to emotional speech. J. Speech Commun. 52, 394–404.

Black, A., Zen, H., Tokuda, K., 2007. Statistical parametric speech synthesis. In: International Conference on Acoustics, Speech, and Signal Processing, pp. 1229–1232.

Chouireb, F., Guerti, M., 2008. Towards a high quality Arabic speech synthesis system based on neural networks and residual excited vocal tract model. Signal Image Video Process. 2, 73–87.

Ciresan, D., Meier, U., Gambardella, L., Schmidhuber, J., 2010. Deep big simple neural nets excel on handwritten digit recognition, CoRR abs/1003.0358.

Clark, R., Richmond, K., King, S., 2007. Multisyn: Open-domain unit selection for the festival speech synthesis system. J. Speech Commun. 49, 317–330.

Deng, L., Yu, D., 2014. Deep Learning: Methods and Applications. MSR-TR-2014-21. NOW Publishers.

Ekpenyong, M., Urua, E., Watts, O., King, S., Yamagishi, J., 2014. Statistical parametric speech synthesis for Ibibio. Speech Commun. 56, 243–251.

Elshafei, M., Al-Muhtaseb, H., Al-Ghamdi, M., 2002. Techniques for high quality Arabic speech synthesis. Inf. Sci. 140, 255–267.

Elshafei, M., Almuhtasib, H., Alghamdi, M., 2006. Machine generation of Arabic diacritical marks. In: The 2006 World Congress in Computer Science Computer Engineering, and Applied Computing, pp. 128–133.

Fares, T., Khalil, A., Hegazy, A., 2008. Usage of the HMM-based speech synthesis for intelligent Arabic voice. In: Int. Conf. Comput. Their Appl, pp. 93–98.

Hamad, M., Hussain, M., 2011. Arabic text-to-speech synthesizer. In: IEEE Student Conference on Research and Development, pp. 409–414.

Hunt, A., Black, A., 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In: International Conference on Acoustics, Speech, and Signal Processing, IEEE, pp. 373–376.

Imai, S., Sumita, K., Furuichi, C., 1983. Mel log spectrum approximation (MLSA) filter for speech synthesis. Trans. IECE of Jpn. 66, 10–18.

Khorsheed, M., 2012. A HMM-based system to diacritize Arabic text. J. Softw. Eng. Appl. 5, 124–127.

Martin, K., Grzl, F., Hannemann, M., Vesely, K., Cernocky, J., 2013. BUT BABEL system for spontaneous cantonese. Interspeech, 2589–2593.

Nose, T., Yamagishi, J., Kobayashi, T., 2005. Acoustic modeling of speaking styles and emotional expressions in HMM-based speech synthesis. IEICE Trans. Inf. Syst. 88, 502–509.

Nose, T., Yamagishi, J., Kobayashi, T., 2007. A style control technique for HMM-based expressive speech synthesis. IEICE Trans. Inf. Syst. E90-D, 1406–1413.

Phan, S., Yu, T., Duong, C., Luong, M., 2013. A study in Vietnamese statistical parametric speech synthesis base on HMM. Int. J. Adv. Comput. Sci. Technol. 2, 1–6.

Qian, Y., Soong, F., Chen, Y., Chu, M., 2006. An HMM-based mandarin Chinese text-to-speech system. In: 5th International Symposium, ISCSLP, pp. 223–232.

Raghavendra, E., Vijayaditya, P., Prahallad, K., 2010. Speech synthesis using artificial neural networks. In: National Conference on Communications, pp. 1–5.

Tokuda, K., Nankaku, Y., Today, T., Zen, H., Yamagishi, J., Oura, K., 2013. Speech synthesis based on hidden Markov models. In: Proceedings of the IEEE, pp. 1234–2125.

Zaki, M., Khalifa, O., Naji, A., 2010. Development of an Arabic text-to-speech system. In: International conference on computer and communication engineering, pp. 1–5.

Zen, H., Senior, A., Schuster, M., 2013. Statistical parametric speech synthesis using deep neural networks. In: International Conference on Acoustics, Speech, and Signal Processing, pp. 7962–7966.

Zen, H., Tokuda, K., Black, A., 2009. Statistical parametric speech synthesis. J. Speech Commun. 51, 1039–1064.