Journal of Computer Science 5 (3): 207-213, 2009
ISSN 1549-3636

An Arabic Text-To-Speech System Based on Artificial Neural Networks

Ghadeer Al-Said and Moussa Abdallah
Department of Electronics Engineering, Princess Sumaya University for Technology, Amman, Jordan

Abstract: Problem statement: With the rapid advancement in information technology and communications, computer systems increasingly offer the users the opportunity to interact with information through speech. The interest in speech synthesis and in building voices is increasing. Worldwide, speech synthesizers have been developed for many popular languages English, Spanish and French and many researches and developments have been applied to those languages. Arabic on the other hand, has been given little attention compared to other languages of similar importance and the research in Arabic is still in its infancy. Based on these ideas, we introduced a system to transform Arabic text that was retrieved from a search engine into spoken words. **Approach:** We designed a text-to-speech system in which we used concatenative speech synthesis approach to synthesize Arabic text. The synthesizer was based on artificial neural networks, specifically the unsupervised learning paradigm. Different sizes of speech units had been used to produce spoken utterances, which are words, diphones and triphones. We also built a dictionary of 500 common words of Arabic. The smaller speech units (diphones and triphones) used for synthesis were chosen to achieve unlimited vocabulary of speech, while the word units were used for synthesizing limited set of sentences. **Results:** The system showed very high accuracy in synthesizing the Arabic text and the output speech was highly intelligible. For the word and diphone unit experiments, we could reach an accuracy of 99% while for the triphone units we reached an accuracy of 86.5%. **Conclusion:** An Arabic text-to-speech synthesizer was built with the ability to produce unlimited number of words with high quality voice.

Key words: Artificial neural networks, text-to-speech synthesis, concatenative synthesis, signal processing

INTRODUCTION

A Text-To-Speech synthesizer (TTS) is a computer-based program in which the system processes through the text and reads it aloud. For most applications, there is a demand on the technology to deliver good and acceptable quality of speech. The quality of a speech synthesizer is judged by its similarity to the human voice (naturalness) and by its ability to be understood (intelligibility). High quality speech synthesis finds a wide range of applications in many fields, to mention a few[1]: Telecommunications services, Language education, Multimedia applications and Aid to handicapped persons.

The speech synthesizer consists of two main components, namely: the text processing component and the Digital Signal Processing (DSP) module. The text processing component has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words, this process is often called text normalization. Then it converts the text into some other representation and output it to the DSP module or synthesizer, which transforms the symbolic information it receives into speech.

The primary technologies for generating synthetic speech waveforms are formant synthesis and concatenative synthesis[2]. Each technology has strengths and weaknesses and the intended uses of a synthesis system will typically determine which approach is used. The speech synthesizer that we built in this work depends on the concatenative synthesis approach. In concatenative synthesis the waveforms are created by concatenating parts of natural speech recorded by humans.

The easiest way to produce intelligible and natural synthetic speech is to concatenate prerecorded utterances. But, this method is limited to one speaker

Corresponding Author: Moussa Abdallah, Department of Electronics Engineering, Princess Sumaya University for Technology, Amman, Jordan

and one voice and the recorded utterances require a larger storing capacity compared to the other methods of speech synthesis. In present systems, the recorded utterances are divided into smaller speech units, such as: words, syllables, phonemes, diphones and sometimes triphones. Word is the most natural unit for the written text and suitable for systems with very limited vocabulary. Diphones are two adjacent half-phones (context-dependent phoneme realizations), cut in the middle and joined into one unit. Triphones are like diphones, but contain one phoneme between steady-state points (half phoneme-phoneme-half phoneme). In other words, a triphone is a phoneme with a specific left and right context[3].

Many researches have been carried out to synthesize speech by different means and for different languages. In 1987, Sejnowsky and Rosenberg constructed a neural network that learns to pronounce English text. The system, which they called NETtalk, was built using a large number of parallel network systems that can capture a significant number of the regularities and many of the irregularities in English pronunciation to convert strings of the English text into strings of phonemes[4]. Some researches used different approaches other than the NETtalk. For example, Karaali *et al*.[5] constructed a rule-based system that uses two neural networks. The first one is a Time-Delay Neural Network to convert a phonetic representation of speech into an acoustic representation and then into speech. The other one is used to control the timing of the output speech.

Regarding the Arabic speech synthesis, Elshafei *et al*.[3] proposed a concatenative Arabic text-to-speech synthesis system that uses diphone/sub-syllable method to construct the spoken utterances. The speech units they used were chosen where the co-articulation effect of the classical Arabic is minimal. They also proposed extension of the set of speech units to improve the quality of the output speech.

MATERIALS AND METHODS

The general architecture of the Text-To-Speech system is shown in Fig. 1. The input to the system is the result of queering an existing search engine which is capable of retrieving Arabic textual data. The text-to-speech synthesis procedure consists of two main phases. The first phase is text analysis. In this phase the input text is pre-processed and then classified using artificial neural networks, we used unsupervised learning paradigm, specifically the kohonen learning rule. Such network can learn to detect the features of the input vector. The second phase is the generation of speech waveforms. Here, we use concatenative speech synthesis approach for this purpose. The post processing is used to smooth the transitions between the concatenated diphones.

The development of a high quality TTS system needs an appropriate database of speech units. Diphones are the main speech units used during the course of this study. The used Arabic diphone database was prepared at "King Abdulaziz City of Science and Technology" in Saudi Arabia, this database contains 368 speech units[6].

Text pre-processing: Before the words enter the neural network, a series of preliminary processing has to be fulfilled. At first, the punctuation marks are removed, then the numbers are identified and the abbreviations are expanded into full words. The next step is to fully diacritise the retrieved text to eliminate any ambiguity about the word's pronunciation. The final step is to prepare the words as input vectors for the neural network. However, neural networks only recognize numerical inputs, therefore, the ASCII code of each character is taken and replaced with its corresponding binary representation. Next the 0's were replaced with (-1)'s to discriminate them from trailing zeros that will be added later. Now the text is ready to be processed and classified by the neural network.

Text to speech conversion: When building a speech synthesizer, one has to decide which synthesis unit to choose. There are different unit sizes and each choice has its own advantages and disadvantages. The longer the unit the more accuracy you get, but at the expense of the number of data needed.

In this research we created three models to handle different sizes of units. These units are words, diphones and triphones, the models will be explained in later sections. In the post processing unit we used three interpolation methods to smooth the transitions between speech units, namely: Linear interpolation, spline interpolation and cubic interpolation.

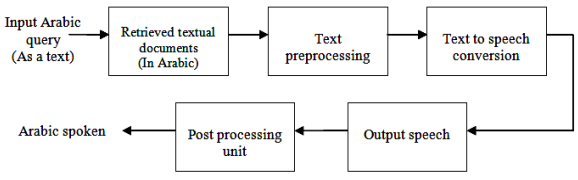


Fig. 1: The basic building blocks of the Arabic TTS

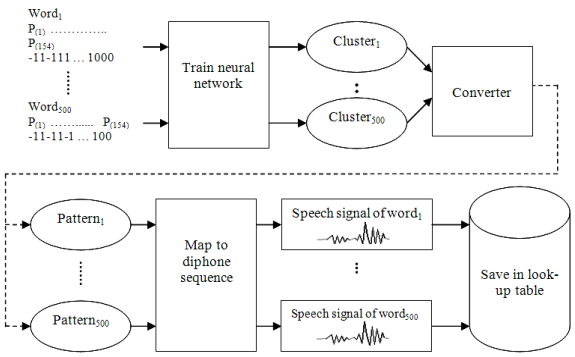


Fig. 2: Word training model

The word model: Systems that simply concatenate isolated words or parts of sentences, are only applicable when a limited vocabulary is required (typically a few hundreds of words) and the sentences to be pronounced respect a very restricted structure[1]. In this model, a dictionary containing 500 words that are commonly used in Arabic is built.

Training the words: The goal of this procedure is to generate the corresponding speech of each word in the dictionary. Since our database of speech doesn't contain complete words, we constructed each word out of its diphone sequence. To train the words of the dictionary, each word is converted into its diphone sequence then passed to the pre-processing unit as explained previously. Neural networks require that all inputs are of the same length, so we chose a vector length of 154 in regard to the longest word in the dictionary. Thus, words producing a vector shorter than 154 are padded with trailing zeros. Figure 2 shows the functional diagram of the training process, the input feature vector is passed to the network at the beginning. The neural network in turn produces a cluster representing the input. Then each cluster is passed to the converter module and is converted into a pattern of 1's and 0's for comparison purposes to be performed later. Now, the pattern is mapped to its corresponding speech signals and saved in a look-up table. This process is performed for all the words of the dictionary.

Synthesizing words: In this process the input text is tokenized into single words and each word is processed individually. Each word goes through the same training process to produce the feature vector and the output pattern. This pattern is then compared with the patterns in look-up table and classified by the Euclidean distance metric. At last, the recognized word is mapped to the corresponding sound and output as a speech. The synthesis procedure is shown in Fig. 3.

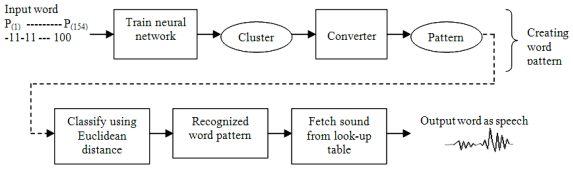


Fig. 3: Word synthesis model

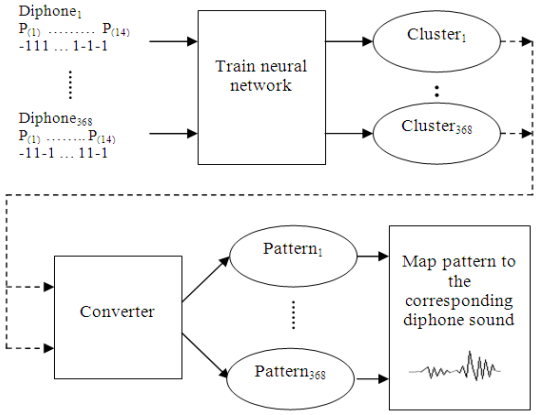


Fig. 4: Diphone training model

If the Euclidean distance exceeds a certain threshold, this means that the word hasn't been recognized as one of the trained words. In this case, it will be added along with its corresponding speech to the look-up table.

The diphone model: We need a more flexible model which adapts to any new input data. Thus, for unrestricted speech synthesis we have to use shorter pieces of speech signal, such as diphones and triphones. The concept of this model is to use the diphones as the speech synthesis units.

Training the diphone database: This step aims to generate a mapping between the textual diphones and their equivalent speech units. Each diphone is represented by two characters, consequently producing a vector of 14 elements. The training process is similar to that of the word model, except that the produced pattern is mapped to the equivalent speech unit of that diphone. This process is repeated for all the diphones in the database. The training process is shown in Fig. 4.

Synthesis using diphone units: To convert input words into speech, the words are automatically broken down to their diphone sequence. Each diphone will be converted into a feature vector then trained by the network to finally produce the pattern. This pattern is classified by the Euclidean distance and the corresponding diphone

speech is fetched. This process is repeated for all the diphones. The output diphone units are saved in a speech buffer until text reading is finished. After that, the speech segments are concatenated together to produce a spoken utterance as shown in Fig. 5.

The triphone model: This model uses longer segmental units (triphones) in attempt to decrease the density of concatenation points, therefore provide better speech quality. The diphones in the speech database were used to build a database of 300 triphones, each triphone is built up by concatenating two diphones. For example the triphone 'Dit' consists of the diphones 'Di' and 'it' connected together. Just note that we built triphones this way provided that the speech units in our hands are diphones, but this is not how triphones are actually constructed.

Training triphones: This procedure is the same as the one used to train diphones, with a difference of the size of the input and output units. A triphone is presented by three characters producing a feature vector of 21 elements. When the pattern is generated, it's mapped to the equivalent triphone speech unit. This process is repeated until the whole 300 triphones are trained.

Synthesis using triphones: To generate spoken utterances in this model, the words are automatically segmented into triphones. These triphones are converted into feature vectors of 21 elements and they go through the same procedure as the diphones, but the output pattern is mapped to the equivalent triphone speech unit. At last, triphone segments are concatenated to produce the written sentence as speech. Figure 6 shows the components of the triphone synthesis system.

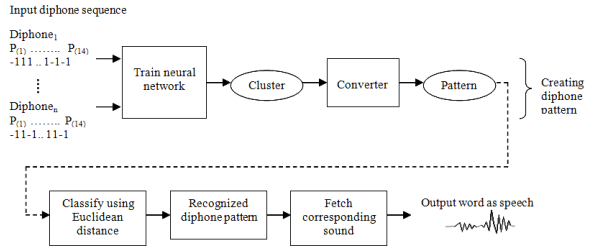


Fig. 5: Diphone synthesis model

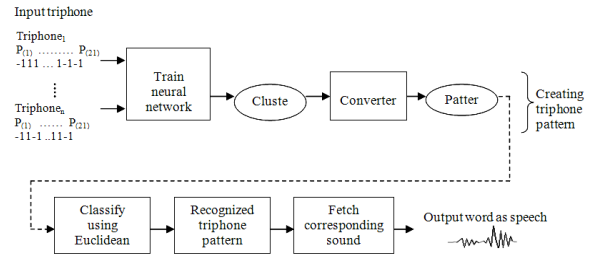


Fig. 6: Triphone synthesis model

RESULTS

The proposed system was built and evaluated using the Matlab7 programming language. To evaluate the accuracy of the synthesizer, different sets of sentences and words are input to the three models (word, diphone and triphone). In order to evaluate the quality of the system, a subjective listening test was conducted. The test sets were played to eight volunteer listeners (4 females and 4 males), which their ages range from 18-34 years. All the listeners are native Arabic speakers and have no experience in listening to synthesized speech. The speech was played by loudspeakers in a quiet room and each listener was tested individually.

As a first step, a set of eight sentences was used to evaluate the word model, in which all the words were recognized by the neural network and output the right speech waveform. Then the output speech was played to the listeners in order to determine how much of the spoken output one could understand, the average of the recognized words by the listeners was 92.26%.

Further, a larger set of sentences was built and tested by the model, but it wasn't evaluated by the listeners. The set contains 27 sentences including the eight sentences tested before. The sentences vary in length from short sentences to a paragraph. The average accuracy of the recognized sentences by the neural network is 99%, Fig. 7 shows the accuracy of each sentence. Finally, to test the whole set of words in the dictionary, the 500 words were input to the neural network in sequence in four different runs. The average accuracy of the four runs is 99.05%.

To evaluate the diphone model, a set of six sentences and nine discrete words where tested both by the neural network and by the listeners. The test conducted by the listeners is the Mean Opinion Score (MOS) test which provides a numerical indication of the perceived quality of received media after compression and/or transmission[7]. The rating scheme is described in Table 1.

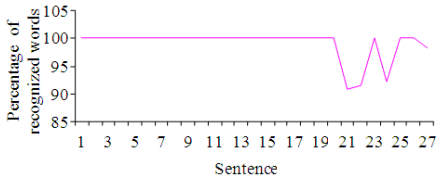


Fig. 7: Accuracy of the word model

Table 1: Mean opinion score rating scheme

Score	Description
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

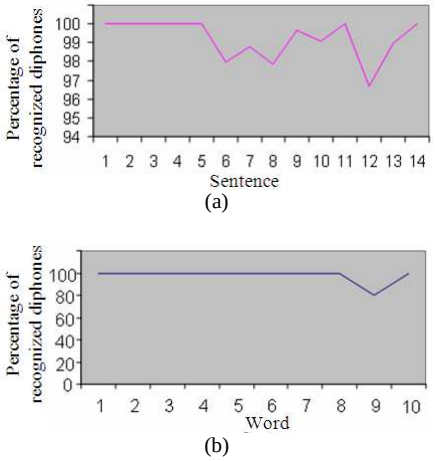


Fig. 8: Diphone recognition accuracy (a): Sentences (b): Discrete words

The accuracy obtained by the neural network was 98% in recognizing the diphones and the average rated score given by the listeners is 4.75.

For further testing, a larger set was created and tested again by this model. The new set consists of fourteen sentences and ten discrete words including the set tested before. The new sentences were also created from words outside the dictionary. The accuracy of the recognized diphones by the neural network is 99.07%. Figure 8 shows diphone recognition accuracy for the new set of sentences and the discrete words.

The same set used to evaluate the diphone model the first time is used to evaluate the triphone model. The recognition accuracy of the six sentences and nine words obtained by the neural network is 86.51%. This result is not as good as the ones obtained by the previous two models. This is due to the small number of triphones in our database, which doesn't cover a wide range of triphone combinations. The triphone recognition accuracy is shown in Fig. 9.

When applying interpolation on the output speech, the results showed that the linear interpolation made no changes on the signal. Meanwhile the spline interpolation did have an effect but it's not the desired one since this kind of interpolation caused the signal to oscillate. The cubic interpolation could successfully smooth the transitions between diphones, but it had a slight effect in improving the quality of the speech when it was played.

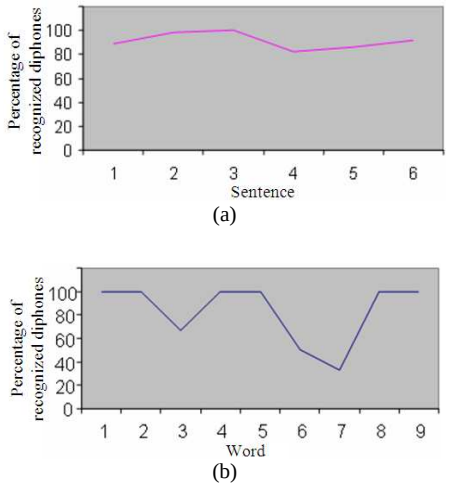
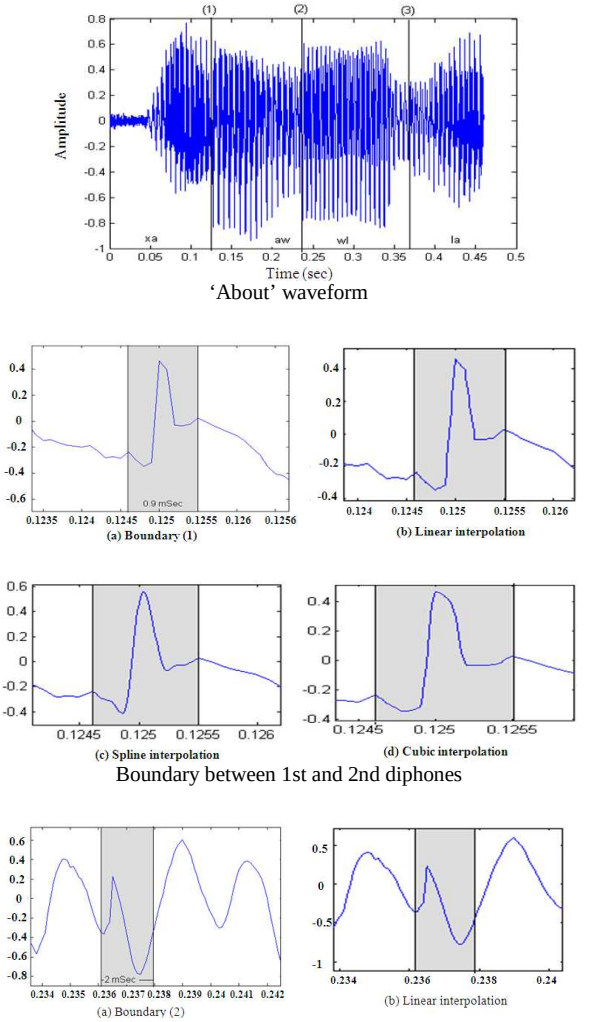


Fig. 9: Triphone recognition accuracy (a): Sentences (b): Discrete words



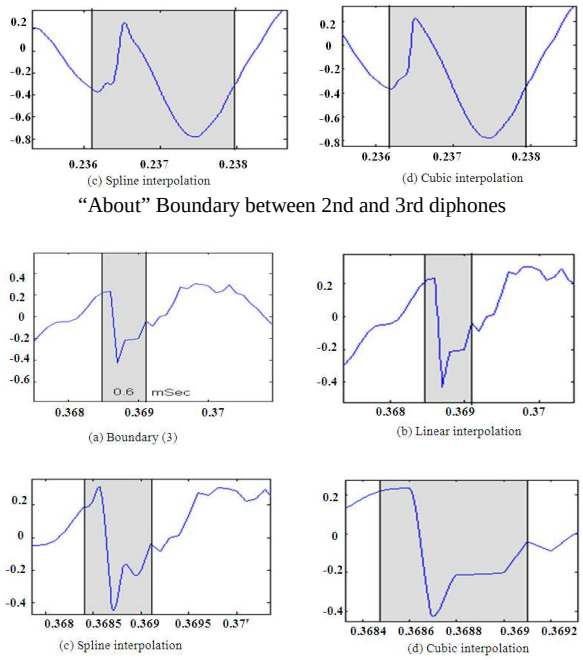


Fig. 10: Interpolating the word "about"

The reason of this shortcoming is the very small duration of the segments we processed where the longest interpolated time span is 2 m sec. which is not adequate to cause a perceptible change in the signal. Figure 10 shows the effect of interpolating the Arabic equivalent of the word "about".

DISCUSSION

In this research, we designed a Text-To-Speech system for synthesizing retrieved Arabic text from different resources, specially the Internet. We used the neural networks with unsupervised learning paradigm, which proved to be a good tool for text to speech synthesis.

Compared to formant synthesis and other technologies, concatenative synthesizers produce more natural speech, but they are usually limited to one speaker and one voice and usually require more memory capacity than other methods. The experiments over the three models we built have shown that words are accurate and fast choice for synthesis, but they require large storage space and only useful for limited vocabulary applications. In our application, diphones showed the best flexibility in building voices over words and triphones, since they produced good quality voice with small number of units compared to the other two models.

The process of concatenating speech units for synthesis causes many cuts in the speech signal, which reinforces the importance of performing some postprocessing to enhance the quality of speech. Among the interpolation methods explained in this study, linear interpolation had no effect in alleviating the sharp transitions, so a smoother interpolating function is desirable. Splines are smooth interpolants but didn't produce the desired improvement in our work, since they caused the signal to oscillate. The cubic interpolation showed better performance compared to linear and spline interpolations. On the other hand, cubic interpolation caused a slight improvement in the speech quality, due to the very short period we performed the interpolation on.

CONCLUSION

In this research, we presented an Arabic text-to-speech synthesis system. Artificial neural networks with unsupervised learning paradigm where used to build the system and different types of speech units were used to synthesize the desired utterances, which are: words, diphones and triphones. The experimental results over the system showed its ability to produce unlimited number of words with high quality voice and high accuracy in converting the written text into speech. Where the obtained accuracy by the word and diphone models was 99% and by the triphone model was 86.5%.

REFERENCES

1. Dutoit, T., 1997. High-quality text-to-speech synthesis: An overview. J. Elect. Electron. Eng. Aust. Special Iss. Speech Recog. Synthes., 17: 25-37. http://www.itcad.net/transcription/IntroTTS.htm
2. Kain Alexander, B. and P.H. Santen Jan Van, 2003. A speech model of acoustic inventories based on asynchronous interpolation. Proceeding of the EUROSPEECH-2003, USA., pp: 329-332. http://www.cslu.ogi.edu/~kain/pub/Kain2003-Eurospeech-AIM.pdf
3. Elshafei, M., H. Al-Muhtaseb and M. Al-Ghamdi, 2002. Techniques for high quality Arabic speech synthesis. Inform. Sci., 140: 255-267. http://www.ccse.kfupm.edu.sa/~elshafei/InformSci e.pdf
4. Sejnowski, T.J. and C.R. Rosenberg, 1987. Parallel networks that learn to pronounce English text. Complex Syst., 1: 145-168. http://www.cnl.salk.edu/ParallelNetsPronounce/ParallelNetsPronounce-TJSejnowski.pdf

5. Karaali, O., G. Corrigan and I. Gerson, 1996. Speech synthesis with neural networks. Proceeding of the World Congress on Neural Networks, San Diego, pp: 45-50. http://arxiv.org/PS_cache/cs/pdf/9811/9811031v1.pdf
6. Alghamdi, M., M. Elshafei and H. Almuhtasib, 2002. Speech units for arabic text-to-speech. Proceeding of the 4th Workshop on Computer and Information Sciences, Dammam, pp: 199-212. http://www.ccse.kfupm.edu.sa/~husni/Research/SpeUniArTexSpe.pdf
7. International Telecommunication Union, 1996. Subjective performance assessment of telephone-band and wideband digital codecs. http://www.itu.int/rec/T-REC-P.83-199303-S/en