Arabic Text-To-Speech Synthesizer

Article · December 2011
DOI: 10.1109/SCoReD.2011.6148774

4 CITATIONS
11 READS

2 authors, including:

Mazin Hamad
University of Khartoum
1 PUBLICATION 4 CITATIONS

Available from: Mazin Hamad
Retrieved on: 19 May 2016
Abstract—Many advances have already been made in Text-To-Speech Synthesizers (TTS). Researchers have been working aggressively on the field of speech synthesis, it is becoming more important everyday and its value prevailed in remedying many development and educational milestones, it is used in many applications and helped users drastically, specially those who need special care and support (such as blind, deafened and vocally handicapped) also speech synthesis help in education and excessive need for computers.

Most of the research on TTS has been made on languages like English and French, while many other languages, like Arabic, have not been taken into consideration until recently and sufficient progress has not been made up to now, thus the area of Arabic text-to-speech systems is still in its early development stages. So the scope of this project is to develop guidelines for Arabic speech synthesis; to touch-up on the kind of challenges might face the scholars down this field when applied to Arabic language, so as to help making the future construction of Arabic voice more promising.

This report presents a workable TTS system for Arabic that uses allophone/diphone concatenation method with two main modules: text/linguistic analyzer and synthesizer core. It takes a complex text as input (including abbreviations, numbers, dates, times, addresses or e-mails) and produces corresponding speech in Arabic. In this system, the output is available in one male voice only. Since Arabic is a verbal language, the developed TTS engine (synthesizer core) can be used also for other verbal languages with some minor modifications involving the phonetic representation of the specific target language together with the preparation and usage of a suitable phonetic database.

Keywords- Text-To-Speech; Arabic; Speech Synthesis; Synthesizer core; Allophone/Diphone Concatenation, Phonetic.

I. INTRODUCTION

Text-to-speech (TTS) synthesis is a research field that has received a lot of attention and resources during the last three decades – for excellent reasons. One of the most interesting ideas (rather futuristic, though) is the fact that a workable TTS system, combined with a workable speech recognition device, would actually be an extremely efficient method for speech coding [1].

Speech synthesis/text-to-speech is an artificial intelligence (AI) science [2], down the track of the Natural Language processing (i.e. it studies the problems of automated generation and understanding of natural human languages).

A TTS system is a system that can convert a given text into speech signals. The source of this text can be very different. While the output of an Optical Character Recognizer (OCR) can be an input for this system, the text that is generated by a language generation system can also be an input for a TTS system.

The aim of an ideal TTS system is to be able to process any text that a human can read (simple or complex). For example, a TTS system should be able to read numbers, to handle abbreviations, to resolve different spellings for a word, etc.

Text-to-speech synthesis systems are an essential component of modern human–machine communication systems and are used to do things like read email messages over a telephone, provide voice output from GPS systems in automobiles, …etc. Another important application is in reading machines for the blind, where an optical character recognition system provides the text input to a speech synthesis system.

Some of existing Arabic TTS products is MBROLA project, Festival Speech Synthesis System (FSST) and the commercial Sakhr TTS engine. Sakhr TTS is the industry leader in synthesizing a natural, human-sounding Arabic voice.

II. CONCATENATIVE SYNTHESIS

One of the most popular methods of synthesizing speech from text is by stringing together or concatenating, prerecorded words, syllables, or other speech segments [3]. This avoids many of the problems encountered in phoneme-to-phoneme synthesis, such as the coarticulatory effect between neighboring speech sounds [4]. Still, even words do not usually occur in isolation: the words immediately preceding or following a given word influence its articulation, its pitch, its duration and stress – often depending on the meaning of the utterance.

Concatenative synthesis (the so called cut and paste synthesis) uses actual short segments of recorded speech that
were cut from recordings and stored in an inventory (“voice database”), either as “waveforms” (uncoded), or encoded by a suitable speech coding method. Synthetic voices are made by concatenating units of sound that have been previously stored in a reference database. The contents of these units and methods of concatenation vary, but the principle of concatenation is universal for TTS involving all but the briefest messages. Nowadays, the use of actual speech waveforms has become increasingly popular, where stored waveforms of various sizes are fetched as needed, with adjustments made mostly at unit boundaries, but sometimes more generally throughout the utterance [5]. Concatenative Synthesis Method uses a large database of source sounds, segmented into units, and a unit selection algorithm that finds the sequence of units that match best the sound or phrase to be synthesized, called the target. The selection is performed according to the descriptors of the units, which are characteristics extracted from the source sounds, or higher-level descriptors attributed to them. The selected units can then be transformed to fully match the target specification, and are concatenated.

A self-explanatory block diagram of a typical concatenative TTS system is shown in Figure 1.

---

### Requirements for a Concatenative Synthesis System

- **Analysis**: source sound files are segmented into units and analyzed to express their characteristics with sound descriptors.

- **Database**: source file references, units and unit descriptors are stored in a database. The subset of the database that is pre-selected for one particular synthesis is called the corpus.

- **Target**: the target specification is generated from a symbolic score (expressed in notes or descriptors), or analyzed from an audio score (using the same segmentation and analysis methods as for the source sounds).

- **Selection**: units are selected from the database that match best the given target descriptors according to a distance function and a concatenation quality function. The selection can be local (the best match for each target unit is found individually), or global (the sequence with the least total distance if found).

- **Synthesis**: is done by concatenation of selected units, possibly applying transformations.

### III. ARABIC TTS SYSTEM MODEL

Figure 2 shows an overview of the developed model for the Arabic TTS system. The input to the program is a plain Arabic text; this text is processed through different consecutive operations, producing a read-out text.

---

Figure 2. Full model of Arabic TTS system.

Processing of the text goes through two main stages, these stages are:

#### A. Text and Linguistic Analysis (High-level Synthesis)

**A.1 Structure Analysis**

Figure 3 depicts the input/output for each sub-module showing each function. This module is responsible for manipulation of both punctuations ("./"،/"؟/"… etc) and special text categories (times, dates, English words, e-mail or website addresses, and numbers).

The input for this module is the whole program input coming either from the user directly through keyboard or from stored text file (.txt). The function of this module is to produce an intermediary text (output) composed only from words (Arabic + English words, e-mails and website addresses) that represent the input for the next stage, namely the Arabic Diacritizer. In addition, the input text is processed to determine where paragraphs start and end, sentences and punctuations (text segmentation).

**A.2 Arabic Diacritizer**

One of the biggest challenges that faces the Arabic text preprocessing is that the text must be diacritized to be read correctly by the synthesizer, so in the preprocessing step each character and its diacritic must be determined. A half-diacritized lexicon of sample Arabic words (diacritization database) is developed using Microsoft Access 2003. The fully diacritized Arabic word can be obtained by passing the retrieved half-diacritized word (obtained from diacritization database) back to the user to add the missing diacritic of the last letter in each word, otherwise the default diacritic "ّ" is assumed and added automatically to the word’s end. It is worth mentioning that all spaces (silence periods) and English words remaining from the previous stage passed this step untouched “English words have no diacritization”. The output of this module includes diacritized Arabic words plus English words from the previous stage in the sequence entered by the user (a FIFO queue is used to reserve the order). The diacritization process is shown in Figure 4.

---

![Diagram](image-url)
B. • Speech (Waveform) Generation (Low-level Synthesis)

B.1 Arabic Grapheme-to-Phoneme conversion module

The Arabic language has about 445 different phonemes which are classified to vowels and constants as shown in Figure 5. In order to assign the closest phoneme (later, sound) to each grapheme taking into consideration that pronunciation - in Arabic - depends on the diacritic, where there are thirteen different diacritic, each of them gives different pronunciation for the letter.

For that reason, sixteen sound files have been recoded for each character and its diacritic. For example: the letter "ف" has the following set of phonemes depending on its diacritic as:

\[
\begin{array}{c}
\text{فِيْ} \\
\text{فﱠ} \\
\text{فﱡ} \\
\text{فﱢ} \\
\text{فْ} \\
\text{ﻓﱠﺎْ} \\
\text{ﻓﱡﻲْ} \\
\text{ﻓﱢﻲْ} \\
\text{فَ} \\
\text{فُ} \\
\text{فِ} \\
\text{فﺎً} \\
\text{فٍ} \\
\text{فٌ} \\
\text{ﻓَﺎْ} \\
\text{ﻓُﻮْ}
\end{array}
\]

Figure 3. Functions of structure analysis module.

Figure 4. Functions of structure analysis module.

Figure 5. Classification charts of Arabic phonemes. [6]
B.2 Arabic Phoneme-to-Sound conversion module
(Arabic Synthesizer)

After Arabic grapheme-to-phoneme conversion phase, all Arabic words are represented, each, by its phonemes corresponding to sequence of wave files that must be read sequentially to synthesize the target Arabic word, the following procedure is used: Since required wave files names are available implicitly (as parts of the phonemic word representation), consecutive file names must be rewritten in an order that is suitable to the sequential wave player. This is achieved by simply entering a back slash (\) between each (.wav) file name and another, and preceding the first (.wav) file name by the absolute path in which the Arabic synthesis database is resident. For example, if the Arabic synthesis database is in folder that has an absolute path name as (C:\ArabicTTS\SynthesisDatabase) and we want the synthesizer to read aloud the word "َقَلَمَ" then the following string should be used for the source file name:

C:\ArabicTTS\SynthesisDatabase\ga,la,mam2.wav

B.3 English speech generation module
(English Synthesizer)

The main target of the project is to synthesize the Arabic text; thus the English synthesizer module is not of much interest. However, since the system also deals with special classes of text that cannot be written in Arabic language such as e-mail, website addresses and abbreviations which are provided in English language; an English TTS engine should be integrated with the Arabic synthesizer. English TTS systems have reached high stages in development compared to the Arabic ones and are integrated in a lot of daily user applications (PDF and text processors). Microsoft SAPI is chosen to be used in the system as the English synthesizer.

B.4 Microsoft Speech API (SAPI)

The Speech Application Programming Interface or SAPI is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications. In general all versions of the API have been designed such that a software developer can write an application to perform speech recognition and synthesis by using a standard set of interfaces, accessible from a variety of programming languages. In addition, it is possible for a 3rd-party company to produce their own Speech Recognition and Text-To-Speech engines or adapt existing engines to work with SAPI. In principle, as long as these engines conform to the defined interfaces they can be used instead of the Microsoft-supplied engines. In addition, the Speech API is a freely-redistributable component which can be shipped with any Windows application that wishes to use speech technology [7]. Microsoft SAPI is used as an English synthesizer.

IV. RESULTS AND DISCUSSION

The program final layout is in the form of Windows-based program that allows the user to go through the steps of language processing and modifying –if needed- in specified sequence and then play the synthesized text. Figure 6 shows the final form of the program window.

The “Testing and Evaluation phase of the developed Arabic TTS System” is done by selecting a test group of random people and allowing them to interact with the program while going through a questionnaire that will be used to evaluate the project throughput. A questionnaire was designed precisely to assess the intelligibility (clearness), naturalness, sound quality and the pronunciation on the level of phoneme word and sentence. The test group was consisting of 20 people of different professions and language knowledge in order to obtain a good assessment (overview of the program's operation).

The samples used in the questionnaire were:

 adultos
HOICE
0 5 4 2 1

A. Intelligibility
A.1 "How much you understand the voice?"

Figure 7. Intelligibility 1st result.
A.2 "Was the voice easy to grab/get?"

Figure 8. Intelligibility 2nd result.

B. • Naturalness

B.1 "Was the sound natural or not?"

Figure 9. Naturalness result.

C. • Sound quality

C.1 "What level of quality do you think the synthesizer has?"

Figure 10. Sound quality result.

D. • Pronunciation

D.1 "Is it very hard to grab/get some of the words?"

Figure 11. Pronunciation 1st result.

D.2 "Did you have to concentrate hard to grab the speech?"

Figure 12. Pronunciation 2nd result.

D.3 "How much annoying did you find the voice?"

Figure 13. Pronunciation 3rd result.
D.4 "Do you think the voice makes many pronunciation mistakes?"

By analyzing results, the intelligibility of the Arabic TTS Synthesizer System was found acceptable; the participants were able to perceive and recognize the majority of sample words and sentences.

Since the concatenated speech is produced from a prerecorded phoneme units; the resultant speech was so natural and of acceptable quality, which can be further enhanced by using longer concatenation units minimizing the large coarticulatory effects that exists between adjacent phones at the cost of degraded naturalness. The synthesizer’s naturalness was not so satisfactory and that can be related back to the difficulties in controlling of the pitch and duration especially when using longer units.

Consonant-Consonant junctions are currently not very satisfactory (large gaps, or sounds get gobbled up).

The speech output appears rather slow, because most of basic sounds have been recorded individually instead of as parts of words. Also, the phonetic description used is cumbersome.

Dynamic speed and pitch modification are not yet incorporated and the synthesizer produces occasional abruptness in case of some words.

Multiple voices need to be added. A major challenge here is to automate the process of going from recording to setting up the database. The main obstacle here is that of automatic segmentation of the recording.

The speech engine does not make any attempt to do prosody effect on the output.

Extensive and continuous work is required to develop the system further and to get a high quality Synthetic speech.

V. CONCLUSIONS

In this report, the development and evaluation of the TTS system for Arabic language based on the concatenation synthesis method is described. The design of an allophone/diphone database and the developed natural language processing modules has been described.

As a result of this project, a working system that gets a plain Arabic text as input and generates the corresponding speech signal for this text is obtained. Although the software part of the system is nearly complete according to aims of the projects, the allophone/diphone database is not perfectly complete yet. However, the results obtained from the current database are quite satisfactory and give an indication that a complete database will match the aims of the project much better.

One point that should be noted is that this is not a high level TTS system; it takes a text as input expecting that the text comes in a phonetic format, i.e. words are in their phonetics forms. Therefore, with some minor refinements, this system can be used to create TTS systems for other languages like English, German, French, etc. with a preprocessing module that converts the text in those languages to the phonetic form that corresponds to this used in the system.

Some other modern techniques that have been applied to speech synthesis, such as Artificial Neural Networks and Hidden Markov Models can be used for the synthesizer core. These methods have been found promising for controlling the synthesizer parameters, such as gain, duration, and fundamental frequency. Also, these methods provide a number of fascinating features including contextual synthesis (prosody support) and trainability, i.e. the quality of the produced speech improves with usage. These models can be applied to Arabic TTS system.

ACKNOWLEDGMENT

The authors would like to thanks their supervisor Dr. Mohammed Ali Hamad Abbas, DEEE, UofK for his kind support, motivation and inspiration that triggered the project. Special thanks are due to Dr. El-Sadig Babiker (Faculty of Arts, UofK, Sudan) for valuable assistance. Also, the authors would like to acknowledge the technical staff at the Electronics Laboratory (UofK, Sudan) for their hospitality and willingness to provide services and facilities under their control.

REFERENCES