CONCATENATIVE ARABIC SPEECH SYNTHESIS USING LARGE SPEECH DATABASE

Wael M. Hamza\(^1\) and Mohsen A. Rashwan\(^2\)

\(^1\)IBM European Speech Research, IBM Egypt, \(^2\)Assoc. Prof., Cairo University.

ABSTRACT

Speech synthesis has got a lot of research interest as it represents an important part in a complete text-to-speech system. In this paper, an Arabic speech synthesis system has been proposed. The proposed system belongs to the family of concatenative speech synthesis systems that use large speech database. The concatenation unit inventory has been automatically constructed from a pre-recorded one hour of speech using context dependent HMM. A unified way to the unit selection has been introduced to enable the use of any type of concatenation units. The introduction of the context cost in the unit selection algorithm makes it easy to use longer and non-uniform units using the same framework. Context cost is represented as the distance between leafs of the context clustering trees that have been grown during the HMM acoustic modeling. Selected unit occurrences have been time and/or pitch scaled to match the required target. This operation is done using an adapted version of sinusoidal model. This version is referred to as Pitch-Synchronous All-Harmonic model. The resulting system has been evaluated using two types of evaluation tests. A word error of 10.3% has been achieved in a DRT-like test while 3.8 score has been recorded in a subjective MOS-like test. These results show that the proposed system can be used as a front-end synthesizer of a complete Arabic text-to-speech system.

1. INTRODUCTION

Text-to-speech (TTS) systems have got a lot of interest recently and a lot of research effort has been invested \([8,1,2]\). This effort is due to usage of the TTS systems in a lot of business, educational and entertainment applications. To ensure high quality output speech, a lot of effort should be invested in three major areas: linguistic analysis of the input text, prosody information generation, and speech synthesis.

In this paper, the third area will be addressed and an Arabic speech synthesizer is proposed. The proposed system belongs to the family of concatenative speech synthesizers that use large speech database \([1,2,5]\). Selecting the concatenation unit type, constructing the unit inventory, selecting between unit occurrences when many are available, and modifying the selected occurrences to match the requested target are four major challenges of concatenative speech synthesis.

In the following sections, all these challenges will be addressed. In Section 2, the unit inventory construction will be described.

2. UNIT INVENTORY CONSTRUCTION

In this section, an automatic procedure to construct the synthesis unit inventory will be described. This is done in three major steps: large database recording, acoustic modeling and database alignment, and occurrence generation.

2.1 Large Database Recording

In this step, a one hour of speech is recorded from the target speaker -the speaker that will be used during the synthesis- using high quality microphone. The recorded script is not phonetically balanced and has been chosen randomly. The recorded speech is sampled at 22KHz sampling rate and quantized at 16-bit/sample. In addition to the speech signal, the laryngograph signal has been simultaneously recorded with the speech signal resulting in a stereo signal that has the speech signal in one channel and the laryngograph signal in the other. The laryngograph signal will be used afterward in determining the glottal closure instant that will be used in the speech modification. The speech signal is then parameterized using the Mel-Frequency Cepstral Coefficients (MFCC). The MFCC vectors are extracted from hamming windowed speech frames of 10 ms with a 5 ms frame shift. As a result of this step, the speech signal, the corresponding text script, the corresponding laryngograph signal and the corresponding MFCC vectors are obtained from the training data.

2.2 Database Alignment

In this step, a three-state left-to-right continuous density Hidden Markov Model (HMM) is constructed for each phone in the phonetic set. Each state contains a single gaussian distribution. Each gaussian is initialized to the global mean and variance of a phonetic set. In this step, a three-state left-to-right continuous density Hidden Markov Model (HMM) is constructed for each phone in the phonetic set. Each state contains a single gaussian distribution. Each gaussian is initialized to the global mean and variance of a phonetic set. In this step, a three-state left-to-right continuous density Hidden Markov Model (HMM) is constructed for each phone in the phonetic set. Each state contains a single gaussian distribution. Each gaussian is initialized to the global mean and variance of a phonetic set. In this step, a three-state left-to-right continuous density Hidden Markov Model (HMM) is constructed for each phone in the phonetic set. Each state contains a single gaussian distribution. Each gaussian is initialized to the global mean and variance of a phonetic set. In this step, a three-state left-to-right continuous density Hidden Markov Model (HMM) is constructed for each phone in the phonetic set. Each state contains a single gaussian distribution. Each gaussian is initialized to the global mean and variance of a phonetic set. In this step, a three-state left-to-right continuous density Hidden Markov Model (HMM) is constructed for each phone in the phonetic set. Each state contains a single gaussian distribution. Each gaussian is initialized to the global mean and variance of a phonetic set. In this step, a three-state left-to-right continuous density Hidden Markov Model (HMM) is constructed for each phone in the phonetic set. Each state contains a single gaussian distribution. Each gaussian is initialized to the global mean and variance of a phonetic set. In this step, a three-state left-to-right continuous density Hidden Markov Model (HMM) is constructed for each phone in the phonetic set. Each state contains a single gaussian distribution. Each gaussian is initialized to the global mean and variance of a phonetic set. In this step, a three-state left-to-right continuous density Hidden Markov Model (HMM) is constructed for each phone in the phonetic set. Each state contains a single gaussian distribution.
Context modeling is a famous problem in speech recognition and a lot of methods exist to solve this problem [4]. In this work, a top-down likelihood-based context-clustering approach [4,5] has been used. The algorithm uses the alignment that results from the previous step to grow a tree for each state in the context independent phone models. A detailed description of the decision-tree-growing algorithm could be found in [5]. As a result of the tree-growing algorithm, each context independent HMM state is replaced with its context dependent version.

Context dependent HMM can then be constructed for the phones by dynamically traversing the resulting context-clustering trees for each state in the given model. These context-dependent models are again trained using the EM algorithm and the resulting models are used to align the training data using the Viterbi algorithm. In this case the resulting alignment is accurate and can be used for inventory construction. The context-clustering tree will be used afterward in the unit selection algorithm as a measure of the context distance.

2.3 Occurrence Generation

As a result of the previous step, an HMM-state based alignment is obtained. Using this alignment, the unit inventory is constructed. Although many unit types could be used, we preferred to use phone as concatenative synthesis unit. For each phone in the phonetic set, all occurrences are stored in the inventory accompanied by some information. This information comprises the location in the speech database, the duration information, and the pitch information. In addition to this, a list of leaf identifications in the context-clustering trees of the phone states are also kept [5]. Starting and ending MFCC vectors are also stored with the occurrence.

3. ONLINE UNIT SELECTION

It is clear from the previous section that, for each unit in the inventory, a number of occurrences is stored for synthesis. The reason for doing this is to make it available to choose the appropriate occurrence that matches the required target. This operation is known as online unit selection and has to be applied in the case of large speech database. This technique was introduced earlier in the CHATR system [3]. The online unit selection algorithm is used here in a unified way to enable the use of any type of synthesis units in the same framework. This is done by incorporating the context cost in the unit selection algorithm. The unit selection algorithm is described as follows. First, the algorithm defines an entity called the target. The target is defined by the set of information that is needed for synthesis. This information comprises the unit identification, the context information, the duration information, and the pitch information. For a sentence to be synthesized, a target string is generated. Then, a cost function between the unit target and the unit occurrence is defined as

\[ C^i(O_t, O_j) = \sum_{j=1}^{n} w_i^j C^j(O_t, O_j) \]  

This cost function is called the target cost. The elements of the target cost functions \( C^i_\)’s are pitch cost, duration cost and the newly introduced context-cost. All these costs are weighted by \( w_i^j \)'s. In addition to this, a cost function between neighboring occurrences is defined as

\[ C^i(O_{t-1}, O_t) = \sum_{j=1}^{n} w_i^j C^j(O_{t-1}, O_t) \]  

This cost is to give more importance to the units occurrences that have smooth concatenation. This is referred to as continuity cost. Continuity cost comprises pitch continuity cost and spectral continuity cost which is calculated using the stored MFCC vectors.

The unit selection algorithm uses dynamic programming to select the unit occurrence sequence that minimizes the cost function

\[ C^i(O_t^*, O_j^*) = \sum_{j=1}^{n} C^i(O_t, O_j) + \sum_{j=2}^{n} C^i(O_{t-1}, O_j) \]  

This dynamic programming algorithm is shown in Figure. 1. In previous work, only occurrences that match the target in context are recalled for the unit selection. This makes it impossible to make use of other occurrences from other contexts even if this context is near the target context. In this work, it is available to choose occurrences from other contexts and incorporating a cost in the target cost function. This cost is represented as the distance between leaves in the context-clustering tree. Although any distance can be used, we have used the Kullback-Leibler distance between the leaf gaussian. This distance is given by [5]:

\[ d = \frac{\delta^i}{\delta} + \frac{\delta}{\delta} + \frac{\delta^i - \mu^i}{\delta} + \frac{\delta^i - \mu^i}{\delta} \]  

where \( \mu \) and \( \delta \) are the mean and the variance of the gaussian and \( i^j \) are the two distribution under distance measure. This cost is used in the cost function as other cost types and is also assigned a weight like all other cost types.

The cost weights are then adjusted using two methods. The first method is to adjust the cost weights manually, while the other one uses linear regression similar to the way used in [3]. Comparable results have been achieved from both methods. The linear regression method is preferred as it does not need time or experience. The major advantage from using the context in the cost function is that one can use longer and non-uniform units in the same online unit selection framework. It also helps to build more detailed contexts because it is allowed to borrow occurrences from another contexts with the appropriate cost. The selected occurrence string is passed to the speech modification/synthesis module for speech production. A detailed description of the unit selection algorithm can be found in [5].
Figure 1: Dynamic programming for unit selection.

4. SPEECH MODIFICATION

Although the resulting occurrence sequence best matches the required target, a step of speech modification is needed to make it exactly match the target. Matching here means that the unit occurrences should have the required pitch and duration information.

A relatively new technique that is based on the commonly known sinusoidal modeling has been proposed in this work. This technique is referred to as Pitch-Synchronous All-Harmonic model. The analysis, modification, and synthesis of this model will be described below.

Analysis: The speech signal is divided into overlapped frames. Each frame is centered at the glottal closure instant in the voiced regions and centered at a constant rate in the unvoiced region. The frame size is four times the local pitch. The speech frame is then multiplied by a blackman window and is passed to the sinusoidal analysis. In the sinusoidal analysis, the speech frame is modeled as a summation of sinusoidal components. The speech frame is represented by:

$$\tilde{z}(n) = \sum_{j=1}^{J_0} A_j \cos(\omega_j n + \phi_j)$$

(5)

where $A_j$, $\omega_j$, and $\phi_j$ are the amplitude, the frequency and the phase of the sinusoidal component $j$ in frame $k$ and $J(k)$ is the number of sinusoids in frame $k$. The sinusoidal analysis should calculate the sinusoidal parameters for the given frame. A detailed description of the calculation of the sinusoidal parameters can be found in [6] and [5] and is briefly described here. The algorithm uses successive approximation to calculate the parameters. This means that one sinusoidal component is calculated after another. The algorithm starts by scanning the frequency ranges from $\omega_0/2$ to $3\omega_0/2$ for the best $\omega$ that minimizes the error between the speech signal and the sinusoidal component that has frequency $\omega$. The search results in the best $\omega$ and the corresponding amplitude and phase. Detailed description of getting the amplitudes and the phases is explained in [6] and [5]. The resulting sinusoidal component is then subtracted form the speech signal, and the operation is repeated for the frequency region between $\omega_0/2$ and $3\omega_0/2$. The operation is performed until the whole band is searched. [6] suggested a way to perform the operation in the frequency domain which reduces the analysis complexity significantly. As a result of the previous steps, a set of frequencies, amplitudes and phases are obtained. The amplitudes are fitted on a cepstral envelope [5]. For each frame, the set of calculated frequencies are used to correct the initial fundamental frequency $\omega_0$ to the fundamental frequency that minimizes the following error

$$E(\omega_0) = \sum_{j=1}^{J_0} |\omega_j - j\omega_0|^2$$

(6)

The sinusoidal representation of the speech frame is then expressed by

$$\tilde{z}(n) = \sum_{j=1}^{J_0} A_j \cos\left[(j\omega_0 + \Delta\omega) n + \phi_j\right]$$

(7)

Modification: The modification is performed in two steps. First, the synthesis pitch marks is derived from the analysis pitch marks depending on the time/pitch modification performed. This is exactly done like the PSOLA method. The calculation of the synthesis pitch marks goes as follows. Given the synthesis pitch mark $t_{i(i)}$, it is required to get the synthesis pitch mark $t_{i(i+1)}$ such that the synthesis pitch at time $t_{i(i)}$ equals to the modified pitch at this time. The algorithm should keep into consideration the time scale warping. This can be done by solving the following integral equation.

$$t_{i(i+1)} - t_{i(i)} = \frac{1}{t_{i(i+1)} - t_{i(i)}} \int_{t_{i(i)}}^{t_{i(i+1)}} \frac{P(t)}{\alpha(t)} dt$$

(8)

where $t_{i(i)}$ and $t_{i(i+1)}$ are the times that correspond to $t_{i(i)}$ and $t_{i(i+1)}$ at the original time axis respectively, $P(t)$ is the original pitch function, and $\alpha(t)$ is the pitch-scale modification factor at time $t$. This equation can be solved by using a piece-wise constants $\alpha$, and $P$ functions. After the determination of the synthesis pitch marks, each synthesis pitch mark is mapped into analysis pitch mark by transforming the synthesis time instants to the original time axis using the time warping function. This operation associates an analysis speech frame with a synthesis pitch mark. For each synthesis frame, the synthesis pitch value, $\omega_{syn}$, is compared to the frame pitch value $\omega_0$, resulting in a pitch modification factor $\beta$. This factor is used to generate the modified speech frame using the modified sinusoidal component. The resulting speech frame can be expressed as

$$\tilde{z}(n) = \sum_{j=1}^{J_0} A_j \cos\left[(j\omega_0 + \Delta\omega) n + \phi_j\right]$$

(9)
where $\beta$ is the pitch modification factor, $A'_{jk}$ and $\phi'_{jk}$ are the new amplitude and phase respectively. The values of $A'_{jk}$ can be obtained by sampling the amplitude envelope at the new pitch harmonics. The values of $\phi'_{jk}$ can be obtained by linearly interpolating the phase values at the original pitch contour. It has to be mentioned that in case of pitch lowering, a high frequency generation step should be performed. High frequency generation can be found in [5]. It has to be mentioned that in unvoiced speech frames, no pitch scale modification is performed and the sinusoidal component parameters are kept unchanged.

**Synthesis:** During the synthesis process, the time domain frame is obtained from the sinusoidal components using equation (9). The resulting short time frames is then overlapped and added using the Least Square synthesis criteria [7].

**5. RESULTS AND DISCUSSION**

To evaluate the output synthesized speech, two types of tests has been developed. First an intelligibility test has been developed based on the Diagnostic Rhythmic Test (DRT) [8]. This test consists of playing back a set of isolated Arabic words and asking the listeners to select from a multiple-choice words the word that best matches the heard word. The words under selection are developed such that they are the same except only one consonant. The words also have the same syllabic structure. The second test is to play a set of twenty sentences and ask the listeners to give a score to the quality of the played sentence. This test is like the Mean Opinion Score test that is used in the field of speech coding and has been used in speech synthesis [1]. Both tests have been performed using a set of thirty listeners that do not work in the field of speech processing. One major problem that faces running these tests, is that there is no prosodic information supplied to the synthesizer. To overcome this problem, the duration of phones in the required utterances has been extracted from naturally spoken utterances. A fixed value pitch contours (monotonic pitch) have been used for the DRT-like test. In the case of the quality test, ten sentences have been synthesized with monotonic pitch while the other ten sentences have been synthesized using natural pitch contours that are extracted from naturally spoken sentences. An average word error of 10.3 % has been achieved in the DRT-like test while a score of 3.8 in the MOS-like test has been recorded. The listeners find the speech output very natural in case of the natural prosody sentence while they found it smooth but robotic in case of monotonic sentences. Tuning the cost weights by hand gives similar results to the linear regression trained weights. There still a lot of effort of having good evaluation tests because subjective tests are very difficult to perform due to the unavailability of the test persons each time the system is tested. Another set of people may give different scores to the same output. Objective tests should be performed to benchmark the progress during the system development.

**6. CONCLUSION**

In the described work a complete Arabic synthesis system is developed. The system represents a promising front-end synthesizer that can be used in a complete Arabic text-to-speech system. The developed system can be the framework in any further research in the field of speech synthesis. It can be also used to test various techniques of prosody generation and other parts of text-to-speech system. The newly introduced context cost makes it possible to use any type of concatenation units in the same framework. Although this system is originally proposed for Arabic speech synthesis, it can be used for other languages. New voices can be built quickly with minimum human effort.

**7. ACKNOWLEDGMENT**

The authors would like to thank RDF on using their speech code base in this work. Their contribution in the listening tests is also appreciated.

**8. REFERENCES**