ELSEVIER

Information Sciences 140 (2002) 255–267

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Techniques for high quality Arabic speech synthesis

Moustafa Elshafei [a], Husni Al-Muhtaseb [a,*], Mansour Al-Ghamdi [b]

[a] *Department of Computer Science, College of Computer Science and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia*
[b] *King Fahd Military College, Riyadh, Saudi Arabia*

Received 1 April 2000; received in revised form 8 May 2001; accepted 28 June 2001

Abstract

The paper proposes a diphone/sub-syllable method for Arabic Text-to-Speech (ATTS) systems. The proposed approach exploits the particular syllabic structure of the Arabic words. For good quality, the boundaries of the speech segments are chosen to occur only at the sustained portion of vowels. The speech segments consists of consonants–half vowels, half vowel–consonants, half vowels, middle portion of vowels, and suffix consonants. The minimum set consists of about 310 segments for classical Arabic. © 2002 Elsevier Science Inc. All rights reserved.

1. Introduction

High quality speech synthesis from electronic form of text has been a focus of research activities during the last two decades, which lead to an increasing horizon of applications. To mention a few, commercial telephone response systems, natural language computer interface, reading machines for blinds and other aids for handicapped, language learning systems, multimedia applications, talking books and toys, and many others.

[*] Corresponding author.
E-mail address: husni@ccse.kfupm.edu.sa (H. Al-Muhtaseb).

PII: S0020-0255(01)00175-X

Fig. 1 shows the functional diagram of the Arabic Text-to-Speech (ATTS) system. It comprises a Natural Language Processing (NLP) engine, capable of producing a phonetic transcription of the text read, together with the desired intonation and stress pattern (prosody), and a Digital Signal Processing (DSP) module, which transforms the symbolic information it receives into speech.

The pre-processing module organizes the input sentences into manageable lists of words or breathing groups. It also identifies numbers, abbreviations, acronyms and idiomatics and transforms them into full text when needed.

The presence of the diacritic marks in the Arabic text is essential for the implementation of the automatic Text-To-Speech (TTS) system. Unfortu-

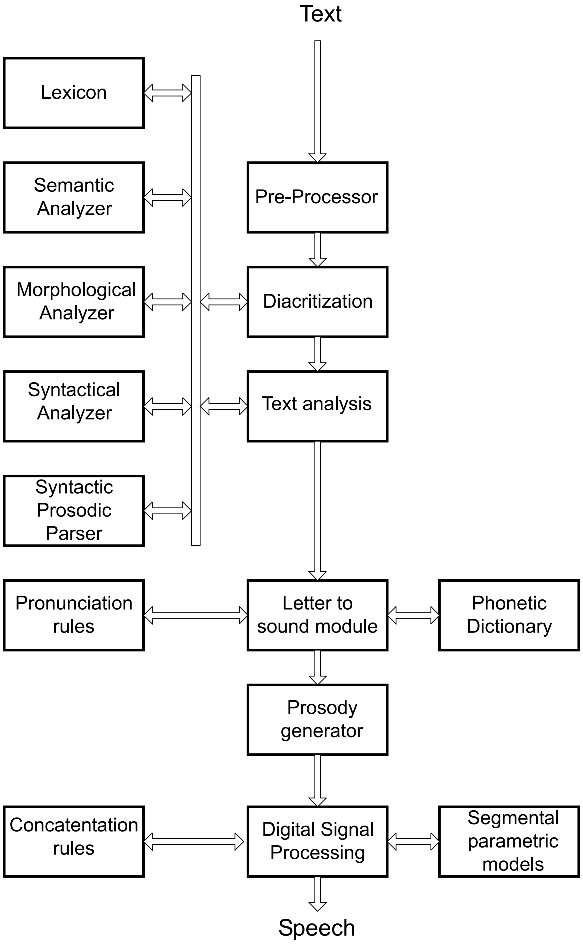


Fig. 1. A general functional diagram of a TTS system.

nately, most modern written Arabic, as in books and newspapers, is at the best partially vowelized. Hence a processor for automatic vowelization of the text must be implemented before applying the TTS rules. The automatic vowelization generator requires integration of morphological, syntactical, and semantic information.

The morphological analysis module proposes all possible part of speech categories for each word taken individually, on the basis of their spelling. Inflected, derived, and compound words are decomposed into their elementary graphemic units (their *morphs*) by simple regular grammars exploiting lexicons of stems and affixes [1].

The NLP engine comprises a morpho-syntactic analyser, underlying the need for some syntactic processing in a high quality TTS system. Indeed, being able to reduce a given sentence into something like the sequence of its parts-of-speech, and to further describe it in the form of a syntax tree, which unveils its internal structure, is required for at least two reasons:

1. Accurate phonetic transcription can only be achieved provided the dependency relationship between successive words is known.
2. Natural prosody heavily relies on syntax. It also obviously has a lot to do with semantics and pragmatics, but since very few data are currently available on the generative aspects of this dependence, TTS systems merely concentrate on syntax. The contextual analysis module considers words in their context, which allows it to reduce the list of their possible part of speech categories to a very restricted number of highly probable hypotheses, given the corresponding possible parts of speech of neighboring words. This can be achieved, for example, by describing local syntactic dependence in the form of probabilistic finite state automata (i.e., as a Markov model), or by *mutli-layer perceptrons* (i.e., neural networks) trained to uncover contextual rewrite rules [2], or with *local, non-stochastic grammars* provided by expert linguists or automatically inferred from a training data set with *Classification And Regression Tree* (CART) techniques [3,4].

Finally, a syntactic–prosodic parser, which examines the remaining search space and finds the text structure (i.e., its organization into clause and phrase-like constituents) which more closely relates to its expected prosodic realization (see below).

The Letter-To-Sound (LTS) module [5] is responsible for the automatic determination of the phonetic transcription of the incoming text. The task of the LTS module is usually classified into *dictionary-based* and *rule-based* strategies, although many intermediate solutions exist. *Dictionary-based* solutions consist of storing a maximum of phonological knowledge into a lexicon. In order to keep its size reasonably small, entries are generally restricted to morphemes, and the pronunciation of surface forms is accounted for by inflectional, derivational, and compounding morphophonemic rules which describe how the phonetic transcriptions of their morphemic constituents are modified when they are combined into words. Morphemes that cannot be found in the lexicon are transcribed by rule [1,2,6].

A rather different strategy is adopted in *rule-based* transcription systems [2], which transfer most of the phonological competence of dictionaries into a set of LTS (or *grapheme-to-phoneme*) rules. This time, only those words that are pronounced in such a particular way that they constitute a rule on their own are stored in an exceptions dictionary. A reasonably small exceptions dictionary can account for a large fraction of the words in a running text. In English, for instance, 2000 words typically suffice to cover 70% of the words in text. Clearly, some trade-off is inescapable. Besides, the compromise is language-dependent, given the obvious differences in the reliability of LTS correspondences for different languages.

2. Segmentation of speech

Connecting prerecorded natural utterances is probably the easiest way to produce intelligible and natural sounding synthetic speech. However, concatenative synthesizers are usually limited to one speaker and one voice and usually require more memory capacity than other methods. One of the most important aspects in concatenative synthesis is to find correct unit length. The selection is usually a trade-off between longer and shorter units. With longer units high naturalness, less concatenation points and good control of coarticulation are achieved, but the amount of required units and memory is increased. With shorter units, less memory is needed, but the sample collecting and labeling procedures become more difficult and complex. In present systems units used are usually words, syllables, demisyllables, phonemes, diphones, and sometimes even triphones.

Word is perhaps the most natural unit for some messaging systems with very limited vocabulary. Concatenation of words is relative easy to perform and coarticulation effects within a word are captured in the stored units. However, there is a great difference with words spoken in isolation and in continuous sentence which makes the continuous speech-to-sound very unnatural [1]. Because there are hundreds of thousands of different words and proper names in each language, word is not a suitable unit for any kind of unrestricted TTS system.

The number of different syllables in each language is considerably smaller than the number of words, but the size of unit database is usually still too large for TTS systems. For example, there are about 10,000 syllables in English. Unlike with words, the inter-syllables coarticulation effect is not included in the stored units. So using syllables as a basic unit is not very reasonable unless various versions of each syllable is also included in the database, which would make the size of the database formidable. There is also no way to control prosodic contours over the sentence. At the moment, no word or syllable based full TTS system exists (English systems).

The current synthesis systems are mostly based on using phonemes, diphones, demisyllables or some kind of combinations of these. Demisyllables represent the initial and final parts of syllables. One advantage of demisyllables is that only about 1000 of them is needed to construct the 10,000 syllables of English [7]. Using demisyllables, instead of for example phonemes and diphones, requires considerably less concatenation points. Demisyllables also take account of most transitions and then also a large number of coarticulation effects and also covers a large number of allophonic variations due to separation of initial and final consonant clusters. However, the memory requirements are still quite high, but tolerable. Compared to phonemes and diphones, the exact number of demisyllables in a language cannot be defined. With purely demisyllable based system, all possible words cannot be synthesized properly [8].

Phonemes are probably the most commonly used units in speech synthesis because they are the normal linguistic presentation of speech. The inventory of basic units is usually between 40 and 50, which is clearly the smallest compared to other units [1]. Each phoneme is represented by a number of templates representing its contextual acoustic variations, called allophones. Using phonemes/allophones gives maximum flexibility with the rule-based systems. However, some phones that do not have a steady-state target position, such as plosives, are difficult to synthesize. The articulation must also be formulated as rules. Phonemes are sometimes used as an input for speech synthesizer to drive for example diphone-based synthesizer.

Diphones (or dyads) are defined to extend the central point of the steady-state part of the phone to the central point of the following one, so they contain the transitions between adjacent phones. This means that the concatenation point will be in the most steady-state region of the signal, which reduces the distortion from concatenation points. Another advantage with diphones is that the coarticulation effect needs no more to be formulated as rules. In principle, the number of diphones is the square of the number of phonemes (plus allophones), but not all combinations of phonemes are needed. The number of units is usually from 1500 to 2000, which increases the memory requirements and makes the data collection more difficult compared to phonemes. However, the number of data is still tolerable and with other advantages, diphone is a very suitable unit for sample-based TTS synthesis. Longer segmental units, such as triphones or tetraphones, are quite rarely used. Triphones are like diphones, but contains 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. For English, more than 10,000 units are required.

Table 1
Features of some popular TTS systems

	System	Technique
1	Festival	Diphone/non-uniform unit selectionn
2	AT&T next generation TTS	Non-uniform unit selection synthesis with half phones as units
3	CHATR ATR-ITL, Kyoto, Japan (English)	Non-uniform unit selection
4	Laureate, British Telecom	Diphone
5	Bell Lab TTS System	LPC diphone synthesis
6	YorkTalk	Non-segmental (formant) synthesis
7	TTS3000 Lernout & Hauspie	Diphone synthesis
8	Pavarobotti, National center for voice and speech	Articulatory speech synthesis (speech and singing)
9	Trainable speech synthesis	HMM-state sized subword uni, Rob Donovan
10	MBROLA, Le Mons Belgien	Diphone synthesis
11	ProVerb, Speech Engine, Elan/CNET	Diphone synthesis
12	Infovox, Telia Promotor, KTH Stockholm	Formant synthesis
13	Multilingual TTS system, TI Uni Duisburg	Formant synthesis
14	DECTalk	A descendent of MITalk and Dennis Klatt's later work with Klattalk

Table 1 gives a brief summary of the main features of some popular commercial and experimental TTS systems.

There are currently several approaches for Arabic speech synthesis; the allophone method [5], the syllable and sub-syllable method [9], and the diphone method [10]. These approaches vary in complexity, memory requirements, and speech quality. The combined Arabic/English allophone set in [5] contains about 150 allophones and consonant–vowel combinations to simplify computer voice production.

3. Diphone database

The basic idea behind building diphone databases is to explicitly list all possible phone–phone transitions in a language. This makes the wrong, but practical, assumption that coarticulatory effects never go over more than two phones. In general the number of diphones in a language is the number of phones squared. In reality, additional sound segments and various allophonic variation may in some cases be also included. The basic idea is to define classes of diphones, for example: vowel–consonant, consonant–vowel, vowel–vowel, and consonant–consonant.

The syllabic structure of Arabic language is exploited here to simplify the required diphones database. The proposed sound segments may be considered as “sub-syllabic” units. For good quality, the diphones boundaries are taken from the middle portion of vowels. Because diphones need to be clearly articulated various techniques have been proposed to extracted them from subjects. One technique uses words within carrier sentences to ensure that the diphones are pronounced with acceptable duration and prosody (i.e., consistent). Ideally, the diphones should come from a middle syllable of nonsense words so it is fully articulated and minimize the articulatory effects at the start and end of the word.

4. Proposed method

The Arabic phoneme set consists basically of 28 consonants, three short vowels, and three long vowels [11]. Several factors affect the pronunciation of phonemes. For example, the position of the phoneme in the syllable as initial, closing, intervocal, or suffix. The pronunciation of consonants may also be influenced by the interaction (coarticulation) with other phonemes in the same syllable. Among these coarticulation effects are the accentuation (pharyngealization) and the nasalization. Arabic vowels are affected as well by the adjacent phonemes. Accordingly, each Arabic vowel has at least three allophones, the normal, the accentuated, and the nasalized allophone.

In classic Arabic, we can divide the Arabic consonants into three categories with respect to dilution and accentuation

- *Always Diluted Consonants* (*ADC*): The Arabic letters Hamza, Baa, Taa, Thaa, Jeem, Hhaa, Daal, Thaal, Zay, Seen, Sheen, Ain, Faa, Kaaf, Meem, Noon, Haa, Waaw (as a consonant), and Yaa (as a consonant) are always diluted. These are 19 consonants.
- *Always Accentuated Consonants* (*AAC*): This group consists of the emphatic phones {Saad /S/, Daad /D/, Zhaa /$\underline{\partial}$/, Thhaa /T/}, and the pharyngeal phonemes {Khaa /x/, Ghain /γ/, and Qaaf /q/}, a total of 7 consonants.
- *Context Dependent Consonants* (*CDC*): The Arabic consonants Raa /r/ and Lam /l/.

A vowel after an Arabic consonant might be diluted or accentuated. The case of a vowel with respect to dilution and accentuation could be specified depending on the predecessor and successor consonants. The following sections elaborate more on these issues.

Arabic language has five syllable patterns: *cV*, *cW*, *cVc*, *cWc* and *cVcc*, where *c* represents a consonant, *V* represents a vowel and *W* represents a long vowel [11].

In the classical Arabic the following rules determine the case of a general vowel with respect to its predecessor/successor consonants within a syllable:

- In cV and cW syllables, a vowel after an *ADC* consonant should be diluted.
- A vowel after an *AAC* consonant should be accentuated.
- A vowel after a *CDC* consonant follows context dependent rules as outlined in [Elshafei].
- A general vowel before an *ADC* consonant may be diluted.
- A general vowel before an *AAC* consonant should be accentuated.

Let us define the *v-elements* to be $c, \underline{c}, x, x', y, y', z, z', v, v', w, w'$, where c represents a consonant, and $\underline{c}$ an intervocal consonant. x is an initial half vowel (HV), x' is an initial accentuated (pharyngealized) HV, y is a final HV, and y' is a final accentuated HV. z represents a sustained portion of a normal vowel, z' is a sustained portion of an accentuated vowel, v is a normal short vowel, v' is an accentuated short vowel, w is a long vowel, and w' is an accentuated long vowel.

Let us also define the *r-phone* set to be $\{cx, cx', y, y', z, z', yc, y'c, y\underline{c}, y'\underline{c}, c, cv, c'v', cw, cw'\}$. The syllable patterns could be mapped to the following patterns:

$$cV : cx\text{–}y, cx'\text{–}y', cv, \text{ or } cv'$$

$$cW : cx\text{–}z\text{–}y, cx'\text{–}z'\text{–}y', cw, \text{ or } cw'$$

$$cVc : cx\text{–}yc, cx'\text{–}y'c, cx\text{–}y'c, \text{ or } cx'\text{–}yc$$

$$cWc : cx\text{–}z\text{–}yc, cx'\text{–}z'\text{–}y'c, cx\text{–}z'\text{–}y'c, \text{ or } cx'\text{–}'z\text{–}yc$$

$$cVcc : cx\text{–}y\underline{c}\text{–}c, cx'\text{–}y'\underline{c}\text{–}c, cx\text{–}y'\underline{c}\text{–}c, \text{ or } cx'\text{–}y\underline{c}\text{–}c$$

A description of these sound patterns and their possible number of segments are discussed below.

Syllable initial consonant – HV (cx). There are 21 consonants that would accept diluted vowels after them. 19 consonants are from *ADC* and two consonants are from *CDC*. Since, we have three different short vowels, this gives us *at most* $21 \times 3 = 63$ possible combinations.

Syllable initial consonant – Accentuated HV (cx'). In classical Arabic, there are nine consonants that would accept accentuated vowels after them. Seven consonants are from *AAC* and two consonants are from *CDC*. Therefore, there are 27 voice segments in this category.

HV – Syllable closing consonant (yc). *At most* we have 63 possible combinations of this form.

HV – Intervocalic consonant ($y\underline{c}$). *At most* we have 63 possible combinations of this form. This is similar to the yc case, except that the consonant here is shorter and unstressed.

Accentuated HV – Syllable closing consonant ($y'c$). This form has *at most* 27 possible combinations.

Accentuated HV – Intervocalic consonant ($y'\underline{c}$). This form has *at most* 27 possible combinations.

Suffix consonant (c). This is the final consonant in the $cVcc$ syllables. The suffix consonants are usually longer than the voweled consonants. The number of such consonants is 28.

Syllable closing HV (y, y'). The number of such units is only 6 combinations.

Steady-state vowels (z, z'). These are sustained vowel sounds of length double the half vowels. There are six of these sustained vowels.

Consonant–short vowels syllables (cv, cv'). There is a total of $28 \times 3 = 84$ syllables. These are redundant and can be composed from cx–y and cx'–y'. However, having them completely stored could lead to a significant improvement in the sound quality.

Consonant–long vowels syllables (cw, cw'). There is a total of $28 \times 3 = 84$ syllables. These are also redundant and can be composed from cx–z–y and cx'–z'–y'. However, having them completely stored could lead to a significant improvement in the sound quality.

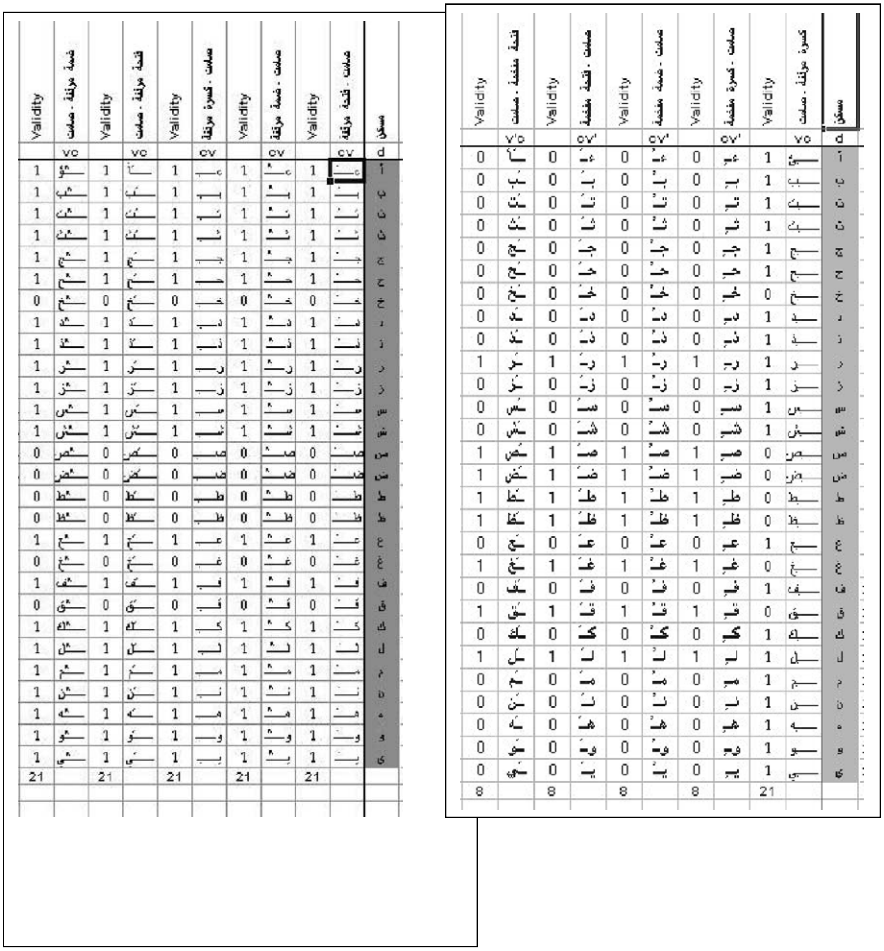
Appendix A shows the detailed combinations of Arabic consonants and vowels. The combinations were divided into seven tables for readability.

5. Implementation issues

1. The minimum set of sound segments consists of the r-phones $\{cx, cx', y, y', z, z', yc, y'c, y\underline{c}, y'\underline{c}, c\}$. The number of these diphones/allphones comes to only 310 segments.
2. About 55% of the Arabic syllables consist of the cV and cW syllables. By adding the extra 168 syllables $\{cv.cv'.cw$, and $cw'\}$ appreciable improvement in the sound quality can be achieved by avoiding concatenation artifacts.
3. In modern standard Arabic significant coarticulation is noticeable. In particular, the effect of the AAC letters on the accentuation of the whole word can be clearly noticed. The effect is both inter-syllabic and intra-syllabic. At the syllable level, in the $\boldsymbol{cWc}$, cVc, and $cVcc$, the entire vowel would be accentuated. To accommodate this effect, the set of sound segments $\{cx', y'c$ and $y'\underline{c}\}$ should be extended to all consonants. The sizes of these sets would become now 84 instead of 27 for the classical Arabic alone.
4. In the $cVcc$ syllables the final consonant cluster is approximated by the xc (or $x'c$) segment and a suffix consonant. That is, in general, a crude approximation. Some consonants in the consonant clusters can be significantly different from normal allphones. However, considering storing ycc and $y'cc$ r-phones can be very costly. it would require about 2352 segments. Fortunately, the $cVcc$ syllable is a rare syllable. it only occurs less that 1% of time. Any reasonable approximation would not lead to severe degradation of the quality.

5. Concatenative synthesizers however make a fixed decision and they cannot reasonably produce anything outside their pre-defined vocabulary. For example, in addition to lacking of foreign phones as *v* and *p*, it would not be able to generate cluster consonants as the word "string" or initial clusters as "cry". This is particularly a problem when it comes to pronouncing foreign names, and terminology. The proposed set needs to be further enhanced with a few English allophones or diphones to be able to accomplish its job.

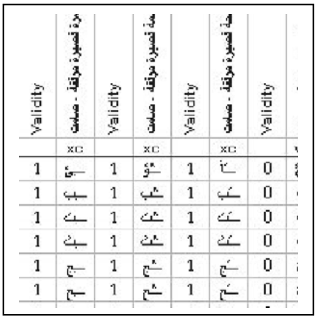
Table 2



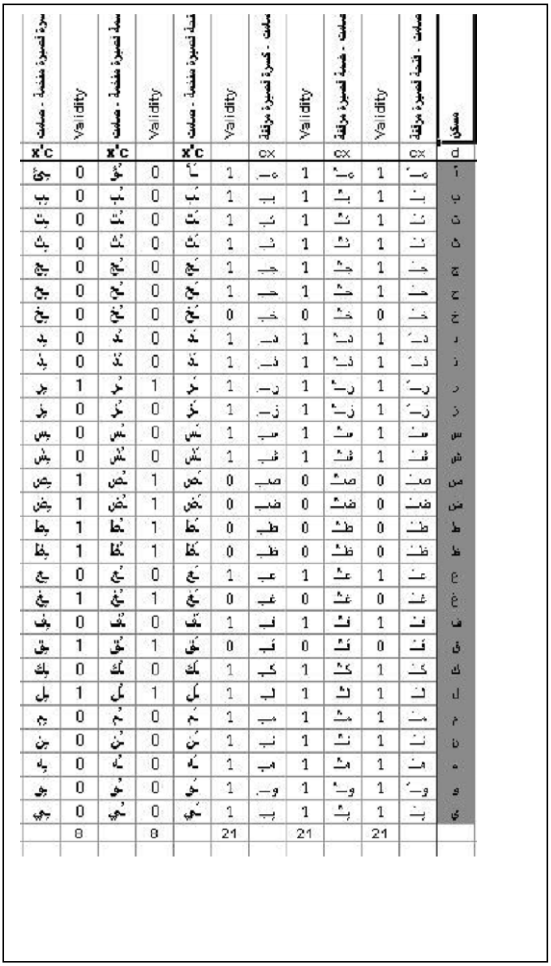
Validity	ضمة مرققة . صامت	Validity	فتحة مرققة . صامت	Validity	صامت . كسرة مرققة	Validity	صامت . ضمة مرققة	Validity	صامت . فتحة مرققة	مسكن
	vc		vc		cv		cv		cv	d
1	ـُؤ	1	ـَأ	1	ءِـ	1	ءُـ	1	ءَـ	أ
1	ـُب	1	ـَب	1	بِـ	1	بُـ	1	بَـ	ب
1	ـُت	1	ـَت	1	تِـ	1	تُـ	1	تَـ	ت
1	ـُث	1	ـَث	1	ثِـ	1	ثُـ	1	ثَـ	ث
1	ـُج	1	ـَج	1	جِـ	1	جُـ	1	جَـ	ج
1	ـُح	1	ـَح	1	حِـ	1	حُـ	1	حَـ	ح
0	ـُخ	0	ـَخ	0	خِـ	0	خُـ	0	خَـ	خ
1	ـُد	1	ـَد	1	دِـ	1	دُـ	1	دَـ	د
1	ـُذ	1	ـَذ	1	ذِـ	1	ذُـ	1	ذَـ	ذ
1	ـُر	1	ـَر	1	رِـ	1	رُـ	1	رَـ	ر
1	ـُز	1	ـَز	1	زِـ	1	زُـ	1	زَـ	ز
1	ـُس	1	ـَس	1	سِـ	1	سُـ	1	سَـ	س
1	ـُش	1	ـَش	1	شِـ	1	شُـ	1	شَـ	ش
0	ـُص	0	ـَص	0	صِـ	0	صُـ	0	صَـ	ص
0	ـُض	0	ـَض	0	ضِـ	0	ضُـ	0	ضَـ	ض
0	ـُط	0	ـَط	0	طِـ	0	طُـ	0	طَـ	ط
0	ـُظ	0	ـَظ	0	ظِـ	0	ظُـ	0	ظَـ	ظ
1	ـُع	1	ـَع	1	عِـ	1	عُـ	1	عَـ	ع
0	ـُغ	0	ـَغ	0	غِـ	0	غُـ	0	غَـ	غ
1	ـُف	1	ـَف	1	فِـ	1	فُـ	1	فَـ	ف
0	ـُق	0	ـَق	0	قِـ	0	قُـ	0	قَـ	ق
1	ـُك	1	ـَك	1	كِـ	1	كُـ	1	كَـ	ك
1	ـُل	1	ـَل	1	لِـ	1	لُـ	1	لَـ	ل
1	ـُم	1	ـَم	1	مِـ	1	مُـ	1	مَـ	م
1	ـُن	1	ـَن	1	نِـ	1	نُـ	1	نَـ	ن
1	ـُه	1	ـَه	1	هِـ	1	هُـ	1	هَـ	ه
1	ـُو	1	ـَو	1	وِـ	1	وُـ	1	وَـ	و
1	ـُي	1	ـَي	1	يِـ	1	يُـ	1	يَـ	ي
21		21		21		21		21		

Validity	فتحة مفخمة . صامت	Validity	صامت . فتحة مفخمة	Validity	صامت . ضمة مفخمة	Validity	صامت . كسرة مفخمة	Validity	كسرة مرققة . صامت	مسكن
	v'c		cv'		cv'		cv'		vc	d
0	ـَأ	0	ءَـ	0	ءُـ	0	ءِـ	1	ـِئ	أ
0	ـَب	0	بَـ	0	بُـ	0	بِـ	1	ـِب	ب
0	ـَت	0	تَـ	0	تُـ	0	تِـ	1	ـِت	ت
0	ـَث	0	ثَـ	0	ثُـ	0	ثِـ	1	ـِث	ث
0	ـَج	0	جَـ	0	جُـ	0	جِـ	1	ـِج	ج
0	ـَح	0	حَـ	0	حُـ	0	حِـ	1	ـِح	ح
0	ـَخ	0	خَـ	0	خُـ	0	خِـ	0	ـِخ	خ
0	ـَد	0	دَـ	0	دُـ	0	دِـ	1	ـِد	د
0	ـَذ	0	ذَـ	0	ذُـ	0	ذِـ	1	ـِذ	ذ
1	ـَر	1	رَـ	1	رُـ	1	رِـ	1	ـِر	ر
0	ـَز	0	زَـ	0	زُـ	0	زِـ	1	ـِز	ز
0	ـَس	0	سَـ	0	سُـ	0	سِـ	1	ـِس	س
0	ـَش	0	شَـ	0	شُـ	0	شِـ	1	ـِش	ش
1	ـَص	1	صَـ	1	صُـ	1	صِـ	0	ـِص	ص
1	ـَض	1	ضَـ	1	ضُـ	1	ضِـ	0	ـِض	ض
1	ـَط	1	طَـ	1	طُـ	1	طِـ	0	ـِط	ط
1	ـَظ	1	ظَـ	1	ظُـ	1	ظِـ	0	ـِظ	ظ
0	ـَع	0	عَـ	0	عُـ	0	عِـ	1	ـِع	ع
1	ـَغ	1	غَـ	1	غُـ	1	غِـ	0	ـِغ	غ
0	ـَف	0	فَـ	0	فُـ	0	فِـ	1	ـِف	ف
1	ـَق	1	قَـ	1	قُـ	1	قِـ	0	ـِق	ق
0	ـَك	0	كَـ	0	كُـ	0	كِـ	1	ـِك	ك
1	ـَل	1	لَـ	1	لُـ	1	لِـ	1	ـِل	ل
0	ـَم	0	مَـ	0	مُـ	0	مِـ	1	ـِم	م
0	ـَن	0	نَـ	0	نُـ	0	نِـ	1	ـِن	ن
0	ـَه	0	هَـ	0	هُـ	0	هِـ	1	ـِه	ه
0	ـَو	0	وَـ	0	وُـ	0	وِـ	1	ـِو	و
0	ـَي	0	يَـ	0	يُـ	0	يِـ	1	ـِي	ي
8		8		8		8		21		

Table 2 (*continued*)



Validity	رة قصيرة مرققة - صامت	Validity	مة قصيرة مرققة - صامت	Validity	حة قصيرة مرققة - صامت	Validity
	xc		xc		xc	
1	ـِئْ	1	ـُؤْ	1	ـَأْ	0
1	ـِبْ	1	ـُبْ	1	ـَبْ	0
1	ـِتْ	1	ـُتْ	1	ـَتْ	0
1	ـِثْ	1	ـُثْ	1	ـَثْ	0
1	ـِجْ	1	ـُجْ	1	ـَجْ	0
1	ـِحْ	1	ـُحْ	1	ـَحْ	0

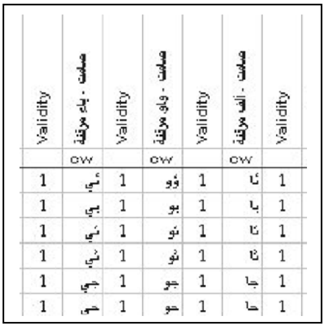


سرة قصيرة مفخمة - صامت	Validity	سمة قصيرة مفخمة - صامت	Validity	فتحة قصيرة مفخمة - صامت	Validity	صامت - كسرة قصيرة مرققة	Validity	صامت - ضمة قصيرة مرققة	Validity	صامت - فتحة قصيرة مرققة	مسكن
x'c		x'c		x'c		cx		cx		cx	d
ـِئْ	0	ـُؤْ	0	ـَأْ	1	ءِـ	1	ءُـ	1	ءَـ	أ
ـِبْ	0	ـُبْ	0	ـَبْ	1	بِـ	1	بُـ	1	بَـ	ب
ـِتْ	0	ـُتْ	0	ـَتْ	1	تِـ	1	تُـ	1	تَـ	ت
ـِثْ	0	ـُثْ	0	ـَثْ	1	ثِـ	1	ثُـ	1	ثَـ	ث
ـِجْ	0	ـُجْ	0	ـَجْ	1	جِـ	1	جُـ	1	جَـ	ج
ـِحْ	0	ـُحْ	0	ـَحْ	1	حِـ	1	حُـ	1	حَـ	ح
ـِخْ	0	ـُخْ	0	ـَخْ	0	خِـ	0	خُـ	0	خَـ	خ
ـِدْ	0	ـُدْ	0	ـَدْ	1	دِـ	1	دُـ	1	دَـ	د
ـِذْ	0	ـُذْ	0	ـَذْ	1	ذِـ	1	ذُـ	1	ذَـ	ذ
ـِرْ	1	ـُرْ	1	ـَرْ	1	رِـ	1	رُـ	1	رَـ	ر
ـِزْ	0	ـُزْ	0	ـَزْ	1	زِـ	1	زُـ	1	زَـ	ز
ـِسْ	0	ـُسْ	0	ـَسْ	1	سِـ	1	سُـ	1	سَـ	س
ـِشْ	0	ـُشْ	0	ـَشْ	1	شِـ	1	شُـ	1	شَـ	ش
ـِصْ	1	ـُصْ	1	ـَصْ	0	صِـ	0	صُـ	0	صَـ	ص
ـِضْ	1	ـُضْ	1	ـَضْ	0	ضِـ	0	ضُـ	0	ضَـ	ض
ـِطْ	1	ـُطْ	1	ـَطْ	0	طِـ	0	طُـ	0	طَـ	ط
ـِظْ	1	ـُظْ	1	ـَظْ	0	ظِـ	0	ظُـ	0	ظَـ	ظ
ـِعْ	0	ـُعْ	0	ـَعْ	1	عِـ	1	عُـ	1	عَـ	ع
ـِغْ	1	ـُغْ	1	ـَغْ	0	غِـ	0	غُـ	0	غَـ	غ
ـِفْ	0	ـُفْ	0	ـَفْ	1	فِـ	1	فُـ	1	فَـ	ف
ـِقْ	1	ـُقْ	1	ـَقْ	0	قِـ	0	قُـ	0	قَـ	ق
ـِكْ	0	ـُكْ	0	ـَكْ	1	كِـ	1	كُـ	1	كَـ	ك
ـِلْ	1	ـُلْ	1	ـَلْ	1	لِـ	1	لُـ	1	لَـ	ل
ـِمْ	0	ـُمْ	0	ـَمْ	1	مِـ	1	مُـ	1	مَـ	م
ـِنْ	0	ـُنْ	0	ـَنْ	1	نِـ	1	نُـ	1	نَـ	ن
ـِهْ	0	ـُهْ	0	ـَهْ	1	هِـ	1	هُـ	1	هَـ	ه
ـِوْ	0	ـُوْ	0	ـَوْ	1	وِـ	1	وُـ	1	وَـ	و
ـِيْ	0	ـُيْ	0	ـَيْ	1	يِـ	1	يُـ	1	يَـ	ي
	8		8		21		21		21		

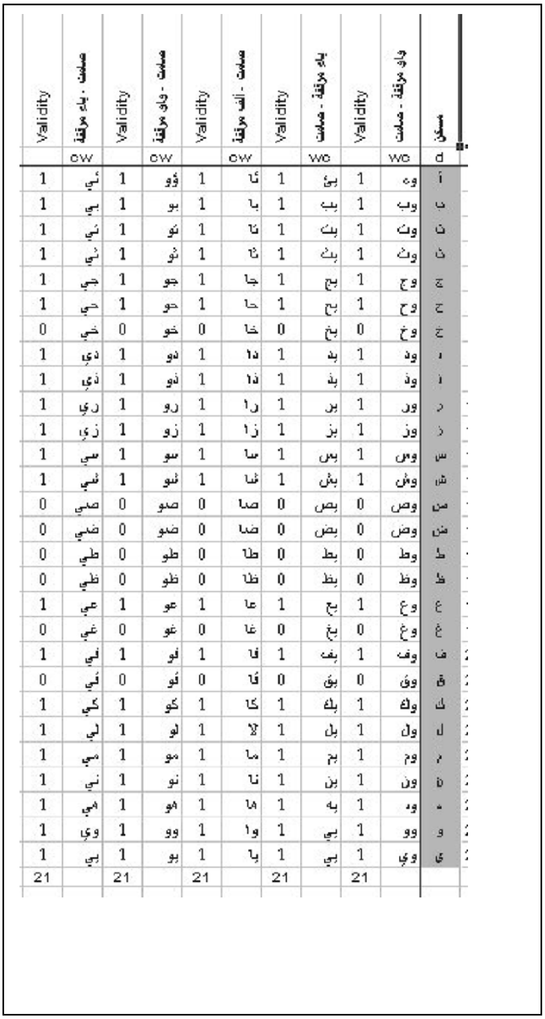
6. Conclusion

The paper defines a set of Arabic diphones/sub-syllables for concatenative ATTS synthesis. The minimum set covers the classical Arabic where the coarticulation is minimal. The paper also proposes extension of the set to improve the quality of the speech and to incorporate the common coarticulation effects found in modern standard Arabic.

Table 2 (*continued*)



Validity	صامت - ياء مرققة	Validity	صامت - واو مرققة	Validity	صامت - ألف مرققة	Validity
	cw		cw		cw	
1	ئي	1	ؤو	1	ئا	1
1	بي	1	بو	1	با	1
1	تي	1	تو	1	تا	1
1	ثي	1	ثو	1	ثا	1
1	جي	1	جو	1	جا	1
1	حي	1	حو	1	حا	1

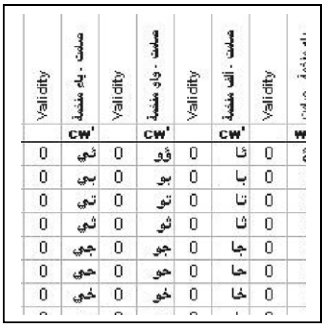


Validity	صامت - ياء مرققة	Validity	صامت - واو مرققة	Validity	صامت - ألف مرققة	Validity	ياء مرققة - صامت	Validity	واو مرققة - صامت	مسكن
	cw		cw		cw		wc		wc	d
1	ئي	1	ؤو	1	ئا	1	بئ	1	وء	أ
1	بي	1	بو	1	با	1	بب	1	وب	ب
1	تي	1	تو	1	تا	1	بت	1	وت	ت
1	ثي	1	ثو	1	ثا	1	بث	1	وث	ث
1	جي	1	جو	1	جا	1	بج	1	وج	ج
1	حي	1	حو	1	حا	1	بح	1	وح	ح
0	خي	0	خو	0	خا	0	بخ	0	وخ	خ
1	دي	1	دو	1	دا	1	بد	1	ود	د
1	ذي	1	ذو	1	ذا	1	بذ	1	وذ	ذ
1	ري	1	رو	1	را	1	بر	1	ور	ر
1	زي	1	زو	1	زا	1	بز	1	وز	ز
1	سي	1	سو	1	سا	1	بس	1	وس	س
1	شي	1	شو	1	شا	1	بش	1	وش	ش
0	صي	0	صو	0	صا	0	بص	0	وص	ص
0	ضي	0	ضو	0	ضا	0	بض	0	وض	ض
0	طي	0	طو	0	طا	0	بط	0	وط	ط
0	ظي	0	ظو	0	ظا	0	بظ	0	وظ	ظ
1	عي	1	عو	1	عا	1	بع	1	وع	ع
0	غي	0	غو	0	غا	0	بغ	0	وغ	غ
1	في	1	فو	1	فا	1	بف	1	وف	ف
0	قي	0	قو	0	قا	0	بق	0	وق	ق
1	كي	1	كو	1	كا	1	بك	1	وك	ك
1	لي	1	لو	1	لا	1	بل	1	ول	ل
1	مي	1	مو	1	ما	1	بم	1	وم	م
1	ني	1	نو	1	نا	1	بن	1	ون	ن
1	هي	1	هو	1	ها	1	به	1	وه	ه
1	وي	1	وو	1	وا	1	بي	1	وو	و
1	بي	1	بو	1	با	1	بي	1	وي	ي
21		21		21		21		21		

Acknowledgements

This work has been supported by KACST under contract number # AT-18-12. The authors would like to acknowledge also the support of KFUPM.

Table 2 (*continued*)



Validity	صامت - ياء منغمة	Validity	صامت - واو منغمة	Validity	صامت - ألف منغمة	Validity	[illegible]
	cw'		cw'		cw'		W
0	ئي	0	ؤو	0	ئا	0	[illegible]
0	بي	0	بو	0	با	0	
0	تي	0	تو	0	تا	0	
0	ثي	0	ثو	0	ثا	0	
0	جي	0	جو	0	جا	0	
0	حي	0	حو	0	حا	0	
0	خي	0	خو	0	خا	0	

Appendix A

See Table 2.

References

[1] J. Allen, S. Hunnicut, D. Klatt, in: D. Klatt (Ed.), From Text To Speech, The MITTALK System, Cambridge University Press, Cambridge, 1987.

[2] S. Furui, M.M. Sondhi (Eds.), Advances in Speech Signal Processing, Marcel Dekker, New York, 1992.

[3] R. Sproat, J. Hirshberg, D. Yarowsky, A Corpus-based Synthesizer, in: Proceedings of the ICSLP 92, Alberta, pp. 563–566.

[4] D. Yarowsky, Homograph Disambiguation in Speech Synthesis, in: Proceedings, 2nd ESCA/IEEE Workshop on Speech Synthesis, New Paltz, NY, 1994.

[5] M. Elshafei Ahmed, Toward an Arabic text-to-speech system, Arabian J. Sci. Eng. 16 (4) (1991).

[6] S.E. Levinson, J.P. Olive, J.S. Tschirgi, Speech synthesis in telecommunications, IEEE Commun. Mag. (November) (1993) 46–53.

[7] R. Donovan, Trainable speech synthesis, Ph.D. Thesis, Engineering Department, Cambridge University, England, 1996.

[8] T. Dutoit, An Introduction to Text-To-Speech Synthesis, Kluwer Academic Publishers, Dordrecht, 1996.

[9] M.F. Abu Alyazeed, M.M.R. Al-Ghoneimy, M.F. Mohammad, Comparison of Syllable and Sub-syllable Methods for Speech Synthesis, in: Proceedings of the Second Conference on Arabic Computational Linguistics, Kuwait, 1989.

[10] Y.A. El-Imam, Unrestricted vocabulary arabic speech synthesis system, IEEE Trans. Acoustic Speech Signal Process. ASSP-37 (12) (1989) 1829–1845.

[11] I. Anees, Al-Aswat Al-Arabia, Arabic title, Anglo-Egyptian Publisher, Egypt, 1990.