ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Speech and Language xxx (2003) xxx–xxx

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Phonetization of Arabic: rules and algorithms

Yousif A. El-Imam *

Department of Electrical and Computer Engineering, Faculty of Engineering, The University of Sharjah, P.O. Box 27272, Sharjah, United Arab Emirates

Received 17 February 2003; received in revised form 4 June 2003; accepted 14 July 2003

Abstract

One approach to the transcription of written text into sounds (phonetization) is to use a set of well-defined language-dependent rules, which are in most situations augmented by a dictionary of exceptional words that constitute their on rules. The process of transcribing into sounds starts by pre-processing the text into lexical items to which the rules are applicable. The rules can be segregated into phonemic and phonetic rules. Phonemic rules operate on the graphemes to convert them into phonemes. Phonetic rules operate onto the phonemes and convert them into phones or actual sounds. Converting from written text into actual sounds and developing a comprehensive set of rules for any language is marked by several problems that have their origins in the relative lack of correspondence between the spelling of the lexical items and their sound contents. For standard Arabic (SA) these problems are not as severe as they are for English or French but they do exist. This paper presents a detailed investigation into all aspects of the phonetization of SA for the purpose of developing a comprehensive system for letter-to-sound conversion for the standard Arabic language and assessing the quality of the letter-to-sound transcription system. In particular the paper deals with the following issues: (1) investigation of the spelling and other problems of SA writing system and their impact on converting graphemes into phonemes. (2) The development of a comprehensive set of rules to be used in the transcription of graphemes into phonemes and (3) investigations of the important contextual phonetic variations of SA phonemes so as to determine viable variants (phones) of the phonemes. (4) The development of a set of rules to be used in the transcription of phonemes into phones. (5) The formulation of the rules for grapheme to phoneme and the phoneme to phone transcriptions into algorithms that lend themselves to computer-based processing. (6) An objective evaluation of the performance of the process of converting SA text into actual sounds.

* Tel.: +971-6-5050964; fax: +971-6-5585191.
E-mail address: yousif@sharjah.ac.ae.

0885-2308/$ - see front matter © 2003 Published by Elsevier Ltd.
doi:10.1016/S0885-2308(03)00035-4

Phonetization of text is an important component in any natural language processing (NLP) domain that envisages text-to-speech (TTS) conversion and has applications beyond speech synthesis such as acoustic modeling for speech recognition and other natural language processing applications.

1. Introduction

Some languages like Spanish, Finish and Swahili have a more or less direct correspondence between the alphabetical writing and the sound systems used. Such languages are relatively easy to transcribe into sounds by simple language-dependent transcription rules. Other languages, like English or French, have only partial regularities between their spelling and sound systems. The correspondence between the orthographic and the sound systems is not obvious for English and French, for such languages a transcription based on rules alone is a formidable task.

The correspondence between Arabic orthographic and sound systems falls in between the simple (Spanish, Finish and Swahili) and the complex (English and French). The Arabic graphemes are easier to transcribe into sounds by, for example, a set of letter-to-sound rules augmented by a dictionary of exceptions. However, due to coarticulations sounds in natural Arabic speech can have enormous contextual variability. Thus the issue in letter-to-sound conversion of standard Arabic (SA) is not just converting the graphemes to the basic abstract linguistic entities like the phonemes but conversion to the phones which represent the actual sounds of the language. This requires that another set of rules have to be developed to cover the important phonetic variations of the SA.

Phonetization of text is an important component of text-to-speech (TTS) conversion. However, there are many domains to the TTS conversion problem, which can be summarized into:

(1) An acoustic phonetic domain whereby the sound system and the phonetics of the language are studied in details in order to specify the phonemes and the contextual variations of these phonemes. The outcome of this step is an acceptable set of synthesis units to be used in producing synthesized speech of a reasonable quality. For SA, there are complex issues to be addressed here, such as its numerous phonetic variations of which emphasis and pharengealization and their influences on any synthesis units chosen merit special attention.

(2) An natural language processing (NLP) domain, which is concerned with the processing of text to extract segmental and suprasegmental information to be used in synthesizing good quality and possibly naturally sounding speech. Phonetization of text is a component in this domain.

(3) A digital signal processing and speech synthesis domain that deals with digital processing of speech signals and developing of synthesis strategies. The issues here are the development of synthesis strategies such as parametric synthesis by rule (formant synthesis and residual LPC synthesis are examples), time-domain synthesis strategy like the waveform concatenation or the unit selection method. The selected strategy must be consistent with the synthesis units chosen in (a).

(4) A domain that deals with quality assessment of the synthesized speech.

It is impossible to focus on all these domains in a single article on TTS conversion of any given language. Previous work on the speech synthesis of Arabic (El-Imam, 1990, 2001) treated, to a reasonable depth, points (1), (3) and (4). That work barely dealt with the intricacies of converting Arabic text into sounds, which is an integral part of the NLP component.

The present article is an attempt to expose, in some details, the issues of converting Arabic text into phonemic and phonetic entities to be used in speech synthesis or other applications and to assess the quality of the output of this important component of NLP. Phonetization of text and the problems associated with it is a discipline that has been a subject of intensive research for other languages like English and French using the various methods that will be outlined in Section 2. In addition, performance evaluation of the output of this important component of TTS has been conducted independent of the other components of TTS (De Mareuil et al., 1998; Yvon et al., 1998). Research on Arabic speech is relatively new. The contribution of this article, when compared to previous work on Arabic speech synthesis, is the detailed exposition and analysis of all aspects of the letter-to-sound conversion problem of SA. These aspects include the following: (1) the exposition of the problems of the writing system of Arabic. (2) The problems related to segmentation and pre-processing of SA. (3) The derivation of both the phonemic and the phonetic transcription rules. (4) The formulation of the rules into algorithms suitable for computer-based processing and the implementation of the algorithms and (5) the assessment of the outcome of this important component of Arabic NLP.

Phonetization of text is important for speech synthesis and recognition and for other natural language processing applications. In speech synthesis, the rules are used to derive the correspondence between the orthography and the sounds (phonemes and phones or allophones). The sounds can then be used alone or converted into syllables, which are further sub-divided into clusters to be used for the language synthesis. In speech recognition, letter-to-sound rules are used as a way of generating pronunciation variants to enhance the quality of the recognizer and generating pronunciations for new add-on words, which are not in the original vocabulary of the speech recognition system. In natural language processing applications, grapheme-to-phoneme conversion (a component of letter-to-sound conversion system) can be used for educational purposes such as correction of spelling mistakes.

A reasonable solution to the problem of converting from letter-to-sound of SA is to have a system for letter-to-sound conversion comprising three components: (1) text segmentation and pre-processing component to format the input text into well-formed lexical items and to convert abbreviations and symbols, acronyms and numbers into word sequences. (2) A grapheme-to-phoneme transcription component that transcribes graphemes into phonemes sequences and (3) a phonetic transcription component that transcribes phonemes-to-phones or actual sounds. The output of the latter two components is a phonetic transcription of input text. Text segmentation and pre-processing is a front end to any text phonetization system and will be dealt with in Section 3.6, after exposition of the problems related to the Arabic writing system. The phonemic and phonetic transcription components are detailed in Sections 4.1 and 4.2.

2. Letter-to-sound transcription methods

There are three methods that have been used for the letter-to-sound transcription of most languages. They are: (1) dictionary-based methods that rely on storing maximum phonological information about morphemes in a lexicon. (2) Rule-based methods whereby expert linguistic and phonetic knowledge is used to develop a set of letter-to-sound rules supported by a lexicon of exceptions when the rules are not applicable and (3) the relatively newer trained data-driven methods, which are classified into three classes.

2.1. Dictionary-based transcription

Dictionary-based letter-to-sound transcription relies on storing maximum phonological knowledge (including pronunciation of morphemes) in a lexicon. Generally morphemes are used instead of words to reduce the size of the lexicon. The pronunciation of input words is generated from the stored morphemes by complex morphological rules that include inflectional, derivational and compounding of morphophonemic rules, that describe how the phonetic transcription of the morphemic constituents vary when they are combined into words. The dictionary-based method, for English, is the work of Coker (1985), Allen et al. (1987), Coker et al. (1990) and Levinson et al. (1993). For French, a very large dictionary is created by Laporte (1988) and used for letter-to-sound conversion. Practical dictionary-based solutions to the letter-to-sound transcription problem of English have been followed in the MITALK TTS system (Allen et al., 1987) and the AT&T TTS system (Levinson et al., 1993; Sproat, 1998).

The lexical entries in the dictionary can have graphemic, phonetic, syntactic and semantic information. A comprehensive dictionary requires huge computer memory and tedious effort during creation. Its main advantage is that it can be used for other purposes such as sentence tagging and parsing necessary for improving intonation and naturalness of speech synthesizers. It can also have applications in machine translation and speech recognition.

Arabic is an inflected language with the result that a root in Arabic can have many inflected forms (an average of ten to twelve forms is usual). For example, the root "درس" (the act of studying) has 11 forms, four verb and seven noun forms. A comprehensive spelling dictionary of Arabic must take into account all these forms resulting in a huge number of entries. For example, the comprehensive Arabic lexical dictionary, Al Qamous Almuhiet (Al-Fairuz Abadi, 1996), has over 250,000 entries. Fortunately, all these derived forms abide by the Arabic spelling rules. This, in addition to the relative simplicity of the Arabic spelling system and its correspondence with the actual pronunciation of the SA words makes a rule-based transcription system, supported by a dictionary of exceptional words, a viable solution to the Arabic letter-to-sound conversion problem.

2.2. Rule-based transcription

Rule-based transcript systems use a comprehensive set of grapheme-to-phoneme rules, a dictionary of exceptions (words that constitute their own rules. The categories of exceptional words in SA are described in Section 4.1), and a phonetic post-processor to transcribe text into actual sounds. Since the emergence of rule-based methods progressive elaborate efforts have been made to design sets of rules and exceptions of wide coverage. For English and French this is done by Ainsworth (1973), McIlroy (1974), Elovitz et al. (1976), Hertz (1979), Hunnicutt (1980), Belrhali et al. (1992) and Divay and Vitale (1997). The most elaborate rule-based systems are expert knowledge-based systems because they use expert linguistic and phonetic knowledge to devise the rules. The different types of rule formalisms are related to the following aspects: differences in number of rules, the phonemic inventory, the types and formats of the rules, the direction in which the rules are parsed, the size of the exceptions' dictionary, the algorithm used to scan the exceptions' dictionary, etc. Rule-based methods are language specific and are widely used in speech synthesis.

The dictionary-based and the rule-based methods are not mutually exclusive. The dictionary of exceptions used in rule-based transcription is a dictionary. The difference between the two methods lies in the relative emphasis when it comes to placing the phonological intelligence. In view of the regular correspondence between the spelling and pronunciation of SA, maximum phonological intelligence is placed on the rules and the Arabic phonetization problem can be handled by a developing a comprehensive set of phonemic and phonetic rules supported by a dictionary of exceptions. On the other hand, languages like English or French have complex spelling systems, which means that developing comprehensive generic context-sensitive phonological rules is almost impossible. This prompts that maximum knowledge is to be placed in the dictionary rather the rules, i.e., the size of the dictionary is large for languages like English or French. Another difference between the dictionary of exceptions, used in rule-based methods and the dictionary as used in dictionary-based transcription systems is in the type of information stored. As pointed earlier, the information in the dictionary of a dictionary-based method is elaborate and can consist of morphemes and their pronunciations and possibly other syntactic and semantic information. In the dictionary of exceptions used in a rule-based method the information stored is exceptional words and their pronunciations. This mixed approach of rules and dictionary has been used for English and is used for Arabic whereby in both cases, the dictionary has to be parsed before execution of the letter-to-sound rules.

Given the nature of the Arabic writing system and the regular relationship between its spelling and pronunciation, it seems that it is well suited to rule-based transcription since it is possible to develop a generalized set of letter-to-sound rules that cover the majority of Arabic spelling. A dictionary of exceptional words will cover the exceptions to the generalized rules. In developing the grapheme-to-phoneme rules, it is assumed that the words are spelled correctly, i.e., vocalized Arabic text is either available or vocalization can be generated by a separate natural language (NLP) component (Elnaggar, 1992, 1993; Elshishini and Elnaggar, 1994; Beesley, 1996, 1998).

Using the spelling literature of SA (Hassanain and Shahata, 1998; Qazzi, 2000; Humoud, 1998), it is possible to compile a set of precise rules to transcribe Arabic text into phonemes and phones. English is a Germanic language that has borrowed heavily from Romance languages. Unlike Arabic, English has a complex spelling system and needs hundreds of letter-to-sound rules to correctly translate about 26% of the words in unlimited text situation as Damper has shown, (Damper et al., 1999) using Elovitz rule set, (Elovitz et al., 1976). For English many frequently used words (like the, of) violate basic pronunciation rules and has to be treated by lists of exceptions. These problems do not exist in Arabic. Putting aside the exceptions, Arabic can be classified as a phonetic language having a regular spelling system. In this respect SA can be modeled with specific rules which are developed manually using our linguistic and phonetic expertise. As will be shown in Sections 4.1 and 4.2, 11 main rules, with their sub-rules, covered the grapheme-to-phoneme conversion and 22 rules covered the phoneme-to-phone components of the letter-to-sound system of SA. Compare these numbers to the 500–4000 rules required for French (De Mareuil et al., 1998). If the list of exceptions is comprehensive and the letter-to-sound rules are complete and precise the transcription of Arabic text would be precise. For Arabic combining precise letter-to-sound rules with a dictionary of exceptional words is enough to achieve high precision in letter-to-sound transcription. Most errors in phonetic transcription are due to some proper nouns and acronyms borrowed from foreign languages. Technically, Arabic is influenced by English and in Arabic foreign words are predominately of English origins, such words are written in the spelling system of the native language and would obey non-Arabic or English letter-to-sound transcription techniques.

2.3. Data-driven transcription

There are three relatively newer data-driven approaches: the pronunciation by analogy (PbA), statistical methods based on stochastic theory and nearest neighbor, and methods based on neural networks. The PbA approach is recently gaining popularity. It exploits the phonological knowledge implicitly contained in a dictionary of words and their corresponding pronunciations. Examples of PbA are the work of Dedina and Nusbaum (1991), Yvon (1996, 1997), Damper and Eastmond (1997), Bagshaw (1998) and Marchand and Damper (2000). Data-driven methods are comprehensively presented in a recent book by Damper (2001). The underlying idea in PbA is to determine the pronunciation of a novel word from similar parts of known words and their corresponding pronunciations. Thus the pronunciation of a novel unknown word is assembled by matching substrings of the input novel word to strings of known lexical words in the dictionary. A partial pronunciation is hypothesized for each matched substring from the phonological knowledge, and the partial pronunciations are concatenated. PbA require a dictionary in which the orthographic forms of the substrings of the words in the dictionary are aligned with the corresponding pronunciations so those matching substrings are easily identified. The statistical methods such as stochastic transduction method (Luk and Damper, 1996), the nearest neighbor approach (Daelemans et al., 1997) is a trained data-driven method. In the IB1-IG method (Daelemans et al., 1997), for example, a training material like the grapheme–phoneme correspondence is used to generate a phoneme classification with a certain probability. Trained neural networks using multilayer perceptrons (MLP) and back-propagation for training have also been used in text transcription such as those developed by Sejnowski and Rosenberg (1987) and Matsumuto and Yamaguchi (1990). MLP-based solutions are language independent, but they have some disadvantages such as the handling of grapheme clusters and syntactic features (Dutoit, 1997, Chapter 5). There are other hybrid approaches (data-driven and rule-based) such as the work done by Meng (Meng, 1995) which can be used either for letter-to-sound as for sound-to-letter conversions. This can be used in both speech synthesis and recognition.

Dictionary-based systems are complex solutions to languages of complex and irregular spelling systems like English and French. The main advantage of dictionary-based and data-driven methods is that they are language independent. However, to introduce these methods into a new language like Arabic, considerable research and manual efforts are needed to create the lexicon and for methods like PbA additional effort is required for aligning the text with the pronunciations. In view of this, the advantages of using dictionary-based transcription or PbA for languages with simpler pronunciation system, like SA have to be carefully weighed against those of rule-based transcription.

3. Grapheme-to-phoneme conversion issues

At the phonemic level, Arabic orthographic transcription is characterized by certain problems that include the following: text normalization, morphophonemic problems, elision, proper names, new and foreign words in the language and spelling irregularities. It should be noted that these problems are common to other languages such as English or French (Divay and Vitale, 1997). For example, spelling problems such as the different phonetic realizations of certain grapheme sequences are encountered in both English and French. Morphophonemic problems such as the dependence of the phonetic realization of a word on the previous and the following words exist in both English and French. Elision is specific to French but to a limited extent it also exists in Arabic. Pronounciation of proper names and neologisms are almost universal problems.

At the phonetic level, the set of rules for transcription from phoneme-to-phones is specific to Arabic. An important component of the phoneme-to-phones transcription is the pharyngealization rules of Arabic (Section 4.2). The input to pharyngealization as well as diphthong generation components of the system, require that the phonemic sequence, which is generated by application of the letter-to-phoneme rules be broken into syllables. This is the process of syllabification of SA phonemes, which is dealt with in Section 3.5.

3.1. The Arabic writing system

The Arabic alphabet has Semitic origins derived from the Aramaic writing system, which is among some of the oldest alphabets in the word (Balabaki, 1981; Sampson, 1985). The Arabic writing system has the regular alphabet for consonants and other diacritics that represent vowels and other symbols used. There are six vowel sounds in Arabic, three short that are phonetically represented by /a/, /u/, /i/ and three long counterparts, which are represented by /a:/, /u:/ and /i:/. There are 28 consonants in Arabic. Using the Speech Assessment Methods Phonetic Alphabet (SAMPA) for Arabic (Gibbon et al., 1997; Department of Phonetic and Linguistics, 2002), the consonant sounds are represented by: ?, b, t, T, g, X\, x, d, D, r, z, s, S, s’, d’, t’, D’, ?’(?\), G, f, q, k, l, m, n, h, w, and j. The Arabic writing system consists of:

(1) Three vowel diacritics that appear on top of the graphemes representing the consonants or on top of Shedda (see item (3)). The vowel diacritics symbolize that the consonant on which they appear is vocalized. For example, the word “جيد” /gajjid/ (good) has the shedda and the “Kasrah” vowel diacritic mark both present on the semivowel /j/.
(2) Twenty-eight graphemes representing the consonant sounds.
(3) The Shedda “w” or gemination sign. The Shedda normally appears on a consonant to indicate that the consonant is geminate or its sound is repeated (see the example in item (1)).
(4) Three Tanween symbols. Tanween Fathah, Tanween Kasrah and Tanween Dummah. They appear on Alef or any consonant to indicate certain phonemic sequences. For example the following words: “كبيرا” /kabi:ran/(enormous), “صغير” /s’aGi:rin/ (small) and “جميل” /gami:-lun/ (beautiful) have the Tanween symbols.
(5) Few ligature symbols like Alef-lam, Lam Alef, etc. For example the words “البيت” /?albajti/ (the house) and “لأن” /la?an/ (because) have the ligature symbols Alef-lam and Lam Alef, respectively.

The vowel symbols are:

- Fathah, a diacritic, which looks like a hyphen, “-”. It appears on top of a consonant to indicate the sound of the Arabic short vowel /a/.
- Kasrah, a diacritic, which looks like a hyphen that appears beneath a consonant “ to indicate that the consonant is vocalized by the sound of the Arabic short vowel /i/.

- Dummah, a diacritic, which looks like a comma that appears on top of a consonant to indicate that the consonant is vocalized by the sound of the short vowel /u/.

The consonant sounds constitute the non-vocalized sounds (when they appear alone, their sounds are non-vocalized) of the Arabic language. In normal speech, the consonants are usually vocalized by the presence of a vowel or a Tanween diacritic on top of their graphemes. Sometimes the consonant sounds are repeated by the presence of a geminating sign "Shedda" on top of their diacritics. The Arabic graphemes, equivalent to the consonant sounds, are, respectively (ء, ب, ت, ث, ج, ح, خ, د, ذ, ر , ز , س, ش, ص, ض, ط, ظ, ع, غ, ف, ق, ك, ل, م, ن, ه, و, and ي). Table 1 shows the phonetic alphabet of Arabic sounds using SAMP for Arabic alphabet. Tables 2 and 3 represent the articulations of the Arabic consonants and vowels sounds.

The Arabic writing system also uses some special symbols and some punctuation marks. One exclusive characteristic of Arabic writing is that the graphemes are connected even when they are printed. An Arabic letter changes its geometrical shape according to its position within the word. In general, there are three shapes for each grapheme and these shapes depend on whether the grapheme appears initial, medial or final in the word. Both the writing and the spelling systems of SA Arabic are uniform throughout the Arabic speaking countries.

3.2. Morphophonemic problems

Like in English and French, the conversion of Arabic graphemes can depend on the preceding and/or following words. In Arabic, this type of context-dependency is encountered with any word

Table 1
The Speech Assessment Methods Phonetic Alphabet (SAMPA) for Arabic

Arabic grapheme	Phonemic symbol	Arabic grapheme	Phonemic symbol
Consonants			
ء	/?/	ض	/d‘/
ب	/b/	ط	/t‘/
ت	/t/	ظ	/D‘/
ث	/T/	ع	/?‘/, /?\/
ج	/g/	غ	/G/
خ	/x/	ف	/f/
ح	/X	ق	/q/
د	/d/	ك	/k/
ذ	/D/	ل	/l/
ر	/r/	م	/m/
ز	/z/	ن	/n/
س	/s/	ه	/h/
ش	/S/	و	/w/
ص	/s‘/	ي	/j/
Vowels		Diphthongs	
…̄.	/a/		/aj/
	/a:/		/aw/
…’.	/i/		
	/i:/		
….̠	/u/		
	/u:/		

Table 2
The standard Arabic consonant phonemes

Place and manner of articulation	Bilabial	Labio-dental	Dental	Alveolar	Post-alveolar	Palatal	Velar	Uvular	Glotal	Pharyngeal
Oral stop	/b/		/d/ /d‘/	/t/ /t‘/			/k/	/q/	/?/	
Nasal (stop)	/m/			/n/						
Affricate						/g/				
Fricative		/f/	/T//D/	/s/ /z/ /s‘//D‘/		/S/	/x/	/G/	/h/	/?‘/,/?\/ /X\/
Lateral				/l/						
Approximant	[w]				/r/	/j/	/w/			

The symbol /w/ is shown in two places in the consonant chart above. This is because it is articulated with both narrowing of the lip aperture, which makes it bilabial, and a raising of the back of the tongue toward the soft palate, which makes it velar. The Arabic /w/ is normally classified as labio-velar semi-vowel.

The phonemes /t‘/, /d‘/, /s‘/ and /D‘/ are the SAMPA symbols for Arabic emphatic characters.

Table 3
Standard Arabic vowel system

Tongue position/height	Front	Central	Back
High or closed	/i/ /i:/ (Unrounded)		
Low or open		/a/ /a:/ (Unrounded)	
High or closed			/u/ /u:/ (Rounded)

that starts with the prefix “ال” (the equivalent of “the” in English) that is followed by what is normally referred to in Arabic by, “Sun letters”. The Arabic alphabet is divided into Shamsi (Sun) and Ghamari (Moon) orthographic characters. The Shamsi characters include coronal sounds that are produced with the tongue blade. The Ghamari characters include non-coronal sounds. The Shamsi characters are: ت, ث, د, ذ, ر, ز, س, ش, ص, ض, ط, ظ, ل, ن and the Ghamari characters are: ء, ا, ب, ج, ح, خ, ع, غ, ف, ق,, ك, م, ه, و, ي. When a word starts with “ال” followed by a “Sun letter” and the word is preceded by a vowel, the prefix “ال” is pronounced as /?a/, the following Sun character is geminated (repeated), and the word that contains the prefix “ال” is merged with its predecessor. For example, “إنكسرت الطاولة” /?inkasarati?at’t’a:wilati/ (the table is broken). If the word that is prefixed with “ال” is not preceded by a vowel, the “ال” is pronounced as /?a/ and the following Sun letter is geminated but the two words are not merged.

3.3. Elision problems

Elision and epenthesis are problems encountered in French (Divay and Vitale, 1997). They are concerned with the pronunciation of the grapheme “e” which is sometimes omitted and becomes an empty phoneme /Φ/ or as the sound of the Schwa. In Arabic, elision problems are encountered with the grapheme “ا” (Alef). The grapheme (Alef) is sometimes omitted or realized as the sound for the long vowel /a:/. If the grapheme “ا”, falls at the end of the word it does not produce a sound and if it falls medial in the word it is used to vocalize the grapheme preceding it by adding a long vowel sound /a:/ to it. For example, in the word “كتبوا” /katabu:/ (they wrote), the grapheme

"اْ" at the end of the word produces no sound and is omitted. In the word "دوَاب" /dawwa:b/ (routinely doing things) the grapheme "اْ", is used to vocalize the grapheme "وَ" "w" to a glide /w/ followed by the sound of the long antral vowel /a:/.

The grapheme "اْ" is also omitted when it appears towards the end of a word before Tanween fataha. For example in the word "كبيراً" /kabi:ran/ (enormous), the Alef is omitted. The grapheme "اْ" can also occur medial as Hamzat Wasl and is omitted. For example, in the word "طأطأ" (nodded his head), the two alefs are Hamzat Wasl and are omitted.

3.4. Proper names and foreign words

Proper names in SA are Arabic or foreign names. Names of Arabic origin obey the spelling rules of Arabic and are handled by the letter-to-sound rules. Examples of proper names in Arabic are: "محمد" / muX\ammad/ (Mohammed), "احمد" /?aX\mad/(Ahmad), etc. A pronunciation dictionary (with appropriate foreign language synthesizer(s)) can handle other foreign words, but one cannot approach exhaustivity. Morphology (Coker et al., 1990) could help, especially with place names. Other schemes use the pronunciations of analogous words (Dedina and Nusbaum, 1991; Damper and Eastmond, 1997; Yvon, 1997; Bagshaw, 1998; and Damper, 2001). In Europe, there are efforts to solve the problem of European proper names using pronunciation dictionaries (Schmidt et al., 1993).

When they appear embedded in a foreign language, foreign proper nouns or words are often capitalized (or italicized) and may obey non-Arabic letter-to-phoneme rules. They can be analyzed for letter sequences unlikely in the Arabic words; then a small set of pronounciation rules (based on an estimate of the identity of the foreign language) could be applied to these words. Vitale and Belhoula, (Vitale, 1991; Belhoula, 1993) developed and utilized such schemes for English, which can as well be used for Arabic.

The problem of neologism or the appearance of new words in SA is caused either by linguistic developments in SA or a dominant foreign language, like English, causing new words to appear or due to technological inventions that prompt the introduction of new words or technical terms. The invention of the car, the airplane, the rocket or the computer and the introduction of these words into the English dictionaries, prompted Arab linguists to find equivalent Arabic words for these English words. The favorable approach used by the Arab linguists was to introduce new Arabic words that stand for the English words on the basis of the functionality of the invented device. For example, the Arabic word used for "computer" is "الحاسب" which is based on its computing functionality. Another, less popular and not favored, approach used by the Arab linguists is to invent an adaptation of any newly introduced foreign word to a pronunciation that matches the structure of Arabic words. For example, the Arabic word used for "television" is "تلفاز" which merely adapts the English pronunciation of the word to a form used in Arabic. Besides, the use of English proper names is also common in Arabic and the pronunciation of these names, in most cases, is kept in its original English form. For such cases we envisage the use of an English TTS system in conjunction with the Arabic TTS to provide partial solution to the problem.

3.5. Syllabification in SA

Syllabification in TTS conversion is important for two reasons. First, it helps the implementation of certain letter-to-phoneme rules such as the diphthong generation in Arabic grapheme-to-phoneme conversion or the treatment of pharengealization in Arabic phoneme-to-phone conversion. Second, syllabification is essential in enhancing the quality of speech produced by synthesizers since detecting the syllable will help in using them to model phone durations and as carriers of certain acoustic traits like intensity and duration to improve the synthesized speech intonation.

In SA there are six types of syllables: CV, CV:, CVC, CV:C and, the rare, CVCC and CV:CC where V stands for a short vowel and V: stands for a long vowel. The first three types are the most common syllable types in SA. Al-Ani, Mitchell and Anis describe the SA syllables (Al-Ani, 1970; Mitchell, 1952; Anis, 1992). The phonemic sequence can be divided into syllables by noting that the nucleus of every SA word is a vowel. This property is very useful in extracting the syllables from a phonemic representation of Arabic words. Every phonemic word is scanned in a backward manner (right-to-left scan starting from the end of the phonemic word) looking for vowels. Whenever a vowel is detected, another more localized scan is performed looking for a syllable match for any of the six syllable types given above. The syllabification algorithm will be presented in Section 5 after the letter-to-phoneme rules are presented in Section 4.

3.6. Text normalization or pre-processing

Text normalization or text pre-processing and formatting is an essential front-end for any system that produces speech sounds from written text. Text pre-processing is needed to prepare the input for further processing and analysis by the remaining modules of the system. Among the tasks allocated to text pre-processing are:

- Text segmentation or the separation of the text into well-formed discrete lexical units such as words.
- Conversions of acronyms, abbreviations and non-alphanumeric characters or symbols into appropriate word or phrase descriptors. For example, the non-alphanumeric character % to the Arabic word sequence, “في المائة” (percent), or the abbreviation “ذ م م” to the Arabic word sequence “ذو مسؤولية محدودة” (limited liability).
- Conversion of dates “21-05-2001” (21/05/2001) into a proper word sequence.
- Conversion of fractional (1/2, 1/4) and whole numbers (256) or numbers made of integer and fraction parts (256.8) into appropriate word sequences.

The pre-processing starts by segmenting the input text into words and sentences. McAllister (McAllister, 1989) proposed dividing the text into orthographic islands (strings of ASCII characters delimited by white space characters, a space, tab or a new-line). To avoid the ambiguities associated with including the punctuation marks into the orthographic islands, the SA punctuation marks are isolated and later associated with the orthographic islands they belong to. This leads to the basic segmentation units. Simple regular rules are then used to convert the basic segmentation units into final segmentation units by examining each incoming basic segmentation unit, in a left-to-right scan. When it comes to sentence end detection, there are punctuation ambiguities, which arise with sentence termination marks (the full stop (.), the question mark (?), the colon (:), the comma (,) and the exclamation mark (!)). The question mark is almost unambiguous because it is usage is on two different levels altogether but the rest have certain ambiguities regarding their usage in text. In showing the time, the colon is used to separate hours from minutes. The exclamation mark is used as a factorial sign in mathematics. For English, the period often appears as part of an abbreviation. In SA, like in English or French, for example, the period and the comma are used in numbers made of integer and fraction parts either to segment a large number (period in French is used for segmenting a large number. In English and SA the comma is used for the same purpose. In French, the comma is used as a decimal point. While in English and SA the period is used for this purpose.

These kinds of ambiguities are context sensitive and are almost universal for all languages that use a system of punctuation similar to English. Liberman and Church (Liberman and Church, 1992) discussed these ambiguities and proposed a probabilistic approach for the entire text segmentation problem, based on tuning a pattern-matching algorithm to data. There are other problems of segmenting text into sentences, for example, quoting direct speech and inserting personal feelings into text. These have received little attention because of their relationship to discourse and pragmatics.

The approach for text segmentation of SA into well-formed sentences follows the proposals discussed above. The basic problems like word tokenization and detection of simple sentence end, word formatting, lexicon lookup and number conversion have been addressed. Some of the ambiguities regarding end of the sentence are context sensitive. For example, the use of the period as a decimal point in numbers made of integer and fraction parts is embedded in between numerals then this can be treated by the component of the pre-processing system that converts numbers to word sequences. The same is done when the comma is used to segment a large number and other ambiguities having specific contexts.

A special word sequence lexicon is used to hold SA abbreviations and symbols, acronyms and words of irregular spellings. Each entry in the lexicon is divided into two fields, the orthography of the item and its representative SA word sequence (for acronyms this can be the pronunciation of the acronym or its long word sequence).

The algorithms for the tokenization of input text, expansion of the abbreviations and symbols, the words or irregular spelling look-up, and number pronunciation are very similar to those used for English and perhaps other languages and will not be explained in this article. However, abbreviations and symbols, acronyms and numbers are included in the performance evaluation of the letter-to-sound system.

4. The general rule formalisms

Generally, the grapheme-to-phoneme rules are not one-to-one. The rules are context-sensitive rewrite rules of similar formalism to Generative Phonology as in SPE (Chomsky and Halle, 1968). The grapheme-to-phoneme rules adopted for the transcription of Arabic text operate on a single level, the grapheme level, and produce an output level containing phoneme. The rules are of the format:

$$A \rightarrow [B]/X\_Y,$$

where A and B can be a single orthographic character, strings of characters, or null. The above rule means that A becomes B if A is in between the left context X and the right context Y. It is important to note that consecutive execution of the rules can lead to potential conflicts if more than one rule is applicable at a certain point. Conflicts occur if the application of a rule produced an output string to which another rule could then apply or if the application of a rule consumes letters in the grapheme level that could otherwise have triggered other rules. It is important to note that the input and output levels of the grapheme-to-phoneme rules are different. The input level contains graphemes and the output level, phonemes. Because the input and output levels are different, the first problem does not arise. The second problem could arise but careful ordering of the rules could alleviate it. In applying the rules, we have found it useful to transform the consonant clusters before the vowels so that their sounds provide additional context information for transforming the vowels.

Rewrite rules can be of multilevel (called multilevel rewrite rules, MLRRs). MLRRs use the left and right contexts of the grapheme under investigation in the input level (in this case, the grapheme level) as well as the values associated with the grapheme in other levels such as the phoneme level. This is done to restrict the application of some rules to a particular sequence of syntactic categories or restrain the application of certain rules by adding contextual information from other levels (Van Leeuwen, 1993; Dutoit, 1997). MLRRs take the form:

$$\mathrm{A} \rightarrow [\mathrm{B}]/\mathrm{X\_Y}/\text{level i}: \mathrm{X}_I\_\mathrm{Y}_{i/}/\text{level j}: \mathrm{X}_j\_\mathrm{Y}_j \ldots /\text{level k}: \mathrm{X}_k\_\mathrm{Y}_k.$$

The above MLLR rule simply states that if A is found in the input level (grapheme level) surrounded by X&Y and by $\mathrm{X}_I\_\mathrm{Y}_I, \mathrm{X}_j\_\mathrm{Y}_j, \ldots, \mathrm{X}_k\_\mathrm{Y}_k$ on other levels $(i, j, \ldots, k)$, the B should be produced on the output level. The other levels could be phoneme or syllabification levels. MLLRs provide solutions to the consumption problem of grapheme as a result of application of certain rules, which we described above. But careful ordering of the rules as noted above could solve this problem. Extracting higher-level features to write MLLRs is not an easy undertaking. Furthermore, the application of MLLRs is complicated by the fact that extra linguistic details are needed such as rule ordering, rule scanning direction (left-to-right or right-to-left) and the scanning hierarchy (from smaller units to bigger units or vice versa (Dutoit, 1997). Nevertheless, MLLR have been successfully applied for grapheme-to-phoneme transcription of complex languages like French. Interestingly enough MLLRs were also used for the transcription of Arabic as part of the MBROLA, Euler project (Dutoit et al., 2000) despite the fact that Arabic is a much simpler language to deal with than French does.

I would have liked to produce a rigorous comparison of the grapheme-to-phoneme rules used for Arabic in the Euler project and the present rules, but nothing have been made public about the Arabic rule set used in the Euler project.

4.1. The phonemic rules of SA

The grapheme-to-phoneme and phoneme-to-phone rules are responsible for the automatic phonetization of written Arabic text. Expert linguistic and phonetic knowledge is used to develop the rules. Grapheme-to-phoneme rules operate on the input Orthography to create the appropriate basic sounds of SA. The rules associate to each sequence of orthographic characters a string of phonemes. However, in speech synthesis and speech recognition, phonetic post-processing is often required to convert the phonemes to a sequence of phones or allophones. The input to this stage is the phonemic string generated by application of the phonemic rules.

The Arabic spelling system has been extensively covered in the literature on Arabic spelling. The most recent contributions to the literature on the Arabic spelling can be found in Hassanain and Shahata (1998), Qazzi (2000) and Humoud (1998). Arabic spelling is fairly regular except in certain situations where certain words violate the regular Arabic pronunciation rules.

The exceptions to the regular SA spelling fall into three predominant categories: (1) violations to the generation of the long vowel /a:/ or Mad Bilalef rule (see rule (9) or “ـَ” (Fathah) → [a:]/X_Y, X = {anygrapheme that represents a consonant} and Y = {“ا”}). The Mad Bilalef rule is violated in numerous words, which are predominantly demonstrative pronouns. Examples are words like “هذا” /ha:Da:/ (this), “هؤلاء” /ha:? ula:/ (those), etc. (2) The abbreviations, the symbols and the acronyms. (3) Some exceptional names like some names of ALLAH (GOD) “الله” /?alla:h/, “الرحمن” /?alraHma:n/ and compounded names derived from them like “عبدالله” /?abd?alla:h/, “عبدالرحمن” //?abd?alraHma:n/. The exceptional words and names are compiled using some of the most famous and comprehensive lexicons of Arabic words and names, Al Qamous Almuhiet (Al-Fairuz Abadi, 1996) and (Humoud, 1995).

The exceptional words are placed into a pronunciation lexicon of exceptions. The phonemic form of each exceptional word is entered against its graphemic form. During text analysis, the exceptions' dictionary is scanned first before the processing of the rules.

For SA, the phonemic rule-set includes 11 categories of grapheme-to-phoneme rules and each category has its own sub-rules. The 11 categories are:

(1) Sukoon deletion rule. In Arabic, “ـْ” (Sukoon) is a silent character, i.e., it does not have any pronunciation and is normally deleted. The Sukoon rule is of the format;

“ـْ” $\rightarrow [\Phi]/X\_Y$,
$X, Y = \{$any grapheme that represents a consonant$\}$,
$\Phi =$ Null phoneme or null grapheme.

(2) Elision rule. There are four sub-rules in this category. The rules apply to the “ا” (Alef) when it occurs final or medial in a word. We have already described these rules in Section 3.3. When a word -final “ا” (Alef) is preceded by “و” “w”, the combination signifies a plural word. In this situation, the final “ا” (Alef) is redundant and does not have any pronunciation. The rule takes the form,

“ا” $\rightarrow [\Phi]/$“و” “w”$\_X$,
$X = \{$any grapheme representing a consonant or it could be null if Alef occurs final in the word$\}$.

For example, in the plural word “كتبوا” (they wrote) the final “ا” is deleted and the word is pronounced as /katabu:/.When the “ا” (Alef), occurs towards the end of the word before final Tanween Fataha, the Alef is omitted. The rule takes the form,

“ا” $\rightarrow [\Phi]/X\_$“ـً”(Tanween Fathah), $X = \{$any grapheme that represents a consonant$\}$.

When the “ا” (Alef), occurs medial as Hamzat Wasl, the Alef is omitted and the Hamazat wasl is pronounced as /?/. The rule takes the form,

“ا” $\rightarrow [?]/X\_$“ء”(Hamza), $X = \{$any grapheme that represents a consonant$\}$.

For example, the word “طأطأ” (he nodded his head). The two Alefs are omitted and the word is pronounced sa t'a?t'a?/.

When the “ا” (Alef), occurs medial in the word it vocalizes the grapheme immediately preceding it by adding the long vowel sound /a:/. The rule takes the form,

“ا” → [a:]/“فتحة”_Y, Y = {any grapheme that represents a consonant}.

For example, in the word “جوَاب” (letter) the “ا” is preceded by “فتحة” and it occurs after the “w”. The “ا” is used to vocalize the “w” by generating the long vowel sound /a:/. The word is pronounced as /gawwa:b/.

(3) Replacement of the grapheme “ى” “j” at the end of the word by the short vowel /a/. This rule is referred to, in Arabic, as: “قانون ياء مقصورة” (Ya Maqsourah) rule.

“ى” “j” → [a]/X_Φ, X = {any grapheme that represents a consonant}.

(“فتحة” is the Arabic name of the grapheme for short vowel /a/. “ضمة” is the Arabic name of the grapheme for short vowel /u/. “كسرة” is the Arabic name of the grapheme for short vowel /i/.) For example, in the word “سلوى” /salwa/ (a female's name), the grapheme “ى” is at the end of the word and is turned into the sound of the short vowel /a/.

(4) Glottal stop insertion rule. There are three sub-rules in this category most of them deal with glottal stop insertion when a word begins, ends, or has a medial “ء” (Hamza). For example, the words “أذهب” /?aDhab/ (go away), “جاء” /ga:?a/ (he came), and “جاءت” /ga:?at/ (she came) begin, ends, and has a medial Hamza. These words will have the glottal stop inserted in their phonetic realizations.

(5) Tanween rules. In Arabic there are certain graphemes, referred to as “Tanween diacritic marks or vocalization symbols”. They are Tanween fatahah “ً”, Tanween damah “ ”, and Tanween Kasrah “ٍ”. They appear at the end of the word. The regular pronunciations of these Tanweens are, respectively, /an/, /un/ and /in/, i.e., one of the three short vowels /a/, /u/ or /i/ followed by the nasal /n/. There are three rules in this category that have the forms,

“ً” → [an]/X_Φ, X = {any grapheme that represent a consonant},

“ً” → [un]/X_Φ, X = {any grapheme that represents a consonant},

“ٍ” → [in]/X_Φ, X = {any grapheme that represents a consonant}.

(6) Gemination or (Shedda) rule. Whenever the germination diacritic “ّ” appears on a character that character is repeated. For example, in the word “جيّد” (good), the germination sign appears above the “يّ” and this causes that character to be repeated and the word is pronounced as /gajjid/. The Shedda rules take the form,

“ّ” → [X]/X_Y,
X = {any grapheme that represents a consonant}, Y = {any grapheme}.

(7) Arabic ligature rules. The Arabic ligatures are a combination of two or more character symbols that are represented in writing by a single orthographic symbol. There are six sub-rules in this category. They apply to “ؤ” (hamza on waw), “ئ” (hamza on yaa), “أ” (hamza on alef), “آ” (alef mada), and the “لأ” (Hamza on Lam Alef). The hamza on waw is pronounced as glottal stop followed by the short vowel /u/. The hamza ya is pronounced as a glottal stop followed by the short vowel /i/. The hamza alef is pronounced as a glottal stop followed by the short vowel /a/. The alef mada is pronounced as a long vowel /a:/, and the Hamza on Lam Alef is pronounced as /l?a?/. The rules take the form,

“ؤ” → [?u]/X_Y, X, Y = {any grapheme that represents a consonant},

“ئ” → [?i]/X_Y, X, Y = {any grapheme that represents a consonant},

“أ” → [?a]/X_Y, X, Y = {any grapheme that represents a consonant},

“آ” → [?a:]/X_Y, X, Y = {any grapheme that represents a consonant},

“لأ” → [la?a]/Φ_Y, Y = {any grapheme that represents a consonant},

“لأ” → [?al?a]/“أ”(Alef)_Y, Y = {any grapheme that represents a consonant}.

The ligature “لا” sometimes appears in certain keyboards and sometimes it can be made from separate “ل” (Lam) and “ا” (Alef). It is usually pronounced as /la:/. It must be noted that there are ligatures made of Shedda and the vowel Symbols (Kasrah, Fatahah or Damah) or Shedda and Tanween symbols. The pronunciations of such ligatures is a serial combination of the pronunciation of the Shedda and the pronunciation of respective vowel or Tanween symbol.

(8) Shamsi rules (applicable to Sun letters) and its counterpart Ghamari rules (applicable to Moon letters) are among the most elaborate in the Arabic orthographic systems. Sun and Moon letters were shown in Section 3.2 when discussing morphophonemic processes. Shamsi and Ghamari rules deal with how to treat the Arabic “ال” (Alef lam) when it occurs before either a Shamsi or a Ghamari character. The “Alef lam” normally pronounced as /?al/, this pronunciation is modified by the presence of a Shamsi character after it. If “Alef lam” occurs before any of the Ghamari characters the default normal pronunciation, /?al/ is used. However, if it occurs before a Shamsi character the story is different and the default pronunciation of the “Alef lam” /?al/ is modified to /?a/, the following Shamsi character becomes geminate or the “ل” (Lam) of the “ال” is assimilated to the sound of the Shamsi character. That is, it is clustered with its own kind to form a geminate CC type of consonant cluster. In addition if the previous word ends with a vowel sound, that word and the word containing the “أل” (Alef lam) are merged together. For example, the word “الشَمس” (the Sun) when spoken in isolation, is pronounced as /?aSSamsu/ while the word “القمر” (the moon) when spoken in isolation, is pronounced as /?alqamaru/. However, when the word “الشَمس” is preceded by a word that ends in a vowel sound like the phrase “طلت الشَمس” (with “كسرة” after the “ت”), the two words are merged into a composite phrase, which is pronounced as /t’ala?’ati?aSamsu/. Since we have 28 Arabic characters, there will be 28 sub-rules in the Shamsi and Ghamari category of rules. The Shamsi rules have the form,

“ال” → [?aYY]/Φ_Y, Y = {any Sun character},

“ال” → [X?aYY]/X_Y,
 X = {any of the vowel graphemes appearing at the end of the previous word},
 Y = {any Sun character}.

The Ghamari rules have the format,

ال→ [?al]/Φ_Y, Y = {any Moon character}.

Sometimes a Hamza is placed on the Alef of Alef lam “ال, such as in “أل”. This hamza on Alef is a Hamzat Wasal that signifies a merger between the current word and its predecessor.

(9) Long vowel generation rules. There are three sub-rules in this category. They are “ـَ” “فتحة” (Fathah) before “ا” generate long vowel /a:/; “ـِ” “كسرة” (Kasrah) before “ي” generate long vowel /i:/ and “ـُ” “ضمَة” (damah) before “و” generate long vowel /u:/. The rules take the form,

“ـَ”(Fathah) → [a:]/X_Y, X = {any grapheme that represents a consonant}, Y = {“ا”},

“ـِ”(Kasrah) → [i:]/X_Y, X = {any grapheme that represents a consonant}, Y = {“ي”},

“ـُ”(damah) → [u:]/X_Y, X = {any grapheme that represents a consonant}, Y = {“و”}.

(10) Diphthong generation rules. There are two sub-rules in this category one for each of the two SA diphthongs. For example, the grapheme sequence ‘فتحة ي’’ “aj” in “جيَد” /gajjid/ (good). The closing syllable grapheme “يj” with the preceding “فتحة” are transformed to the diphthong sound cluster /aj/. The rule takes the form,

“ي” “j” → [aj]/“فتحة”_“ي j, when closing a syllable of type CVC”.

Similar rule applies to the diphthong /aw/. For example, the grapheme sequence “و فتحة” “aw” in the word “جوَاد” /gawwa:d/ (horse). The closing syllable grapheme “w” with the preceding “فتحة” is transformed to the diphthong sound cluster /aj/. The rule takes the form,

“و” “w” → [aw]/“فتحة”_“و w, when closing a syllable of type CVC”.

During implementation of the rules, the application of the diphthong rules is deferred until syllabification is performed. This is important because the end of closing syllable has to be detected before using the diphthong rule to detect the diphthong itself.

(11) Short vowel replacement rules. There are three rules, one for each short vowel grapheme (fataha, damah and Kasrah), in this category. Any of the three short vowel graphemes that are not consumed is replaced by its sound transcription. For example, the grapheme “ـَ” (Fathah) is replace by the vowel phoneme /a/. The “ـِ” (Kasrah) is replaced by /i/. The “ـُ” (damah) is replaced by /u/.

“ـَ”(Fathah) → [a:]/X_Y,
X = {any grapheme that represents a consonant},
Y = {any grapheme that represents a consonant, excluding the“ا”(Alef)}.

“ـِ”(Kasrah) → [i:]/X_Y,
X = {any grapheme that represents a consonant},
Y = {any grapheme that represents a consonant, excluding the“ي” “j”}.

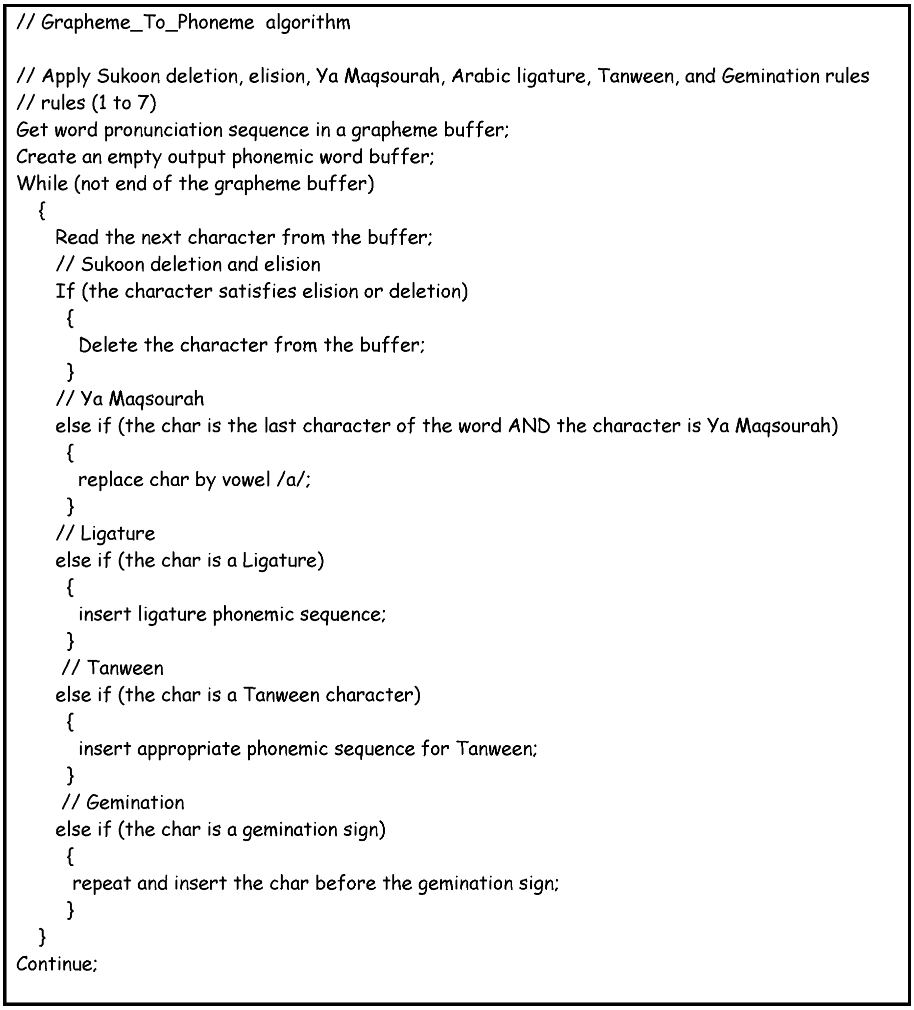
“ـُ”(damah) → [u:]/X_Y,
X = {any grapheme that represents a consonant},
Y = {any grapheme that represents a consonant, excluding“و” “w”}.

The exclusions, mentioned in the Y contexts, are not essential if the long vowel generation rules are applied or scanned before the short vowel generation rules. If this is done, any Fathah preceding “ا”, damah, preceding “و”, or Kasrah preceding “ي” will be consumed by the application of the long vowel generation rules.

The algorithm for realizing the grapheme-to-phoneme rules is presented by the pseudo-code shown in Fig. 1. Comments on the pseudo-code indicate the position where a certain rule(s) is applied.

4.2. The phonetic rules

Additional phoneme-to-phone rules are used to cover the essential phonetic (allophonic) variants of the sounds of Arabic. The important phonetic variations of Arabic sounds include pharyngealization of vowels and diphthongs, nasalization of vowels and diphthongs, and other anticipatory coarticulation like sound overlapping and adaptations.



```
// Grapheme_To_Phoneme  algorithm

// Apply Sukoon deletion, elision, Ya Maqsourah, Arabic ligature, Tanween, and Gemination rules
// rules (1 to 7)
Get word pronunciation sequence in a grapheme buffer;
Create an empty output phonemic word buffer;
While (not end of the grapheme buffer)
  {
    Read the next character from the buffer;
    // Sukoon deletion and elision
    If (the character satisfies elision or deletion)
      {
        Delete the character from the buffer;
      }
    // Ya Maqsourah
    else if (the char is the last character of the word AND the character is Ya Maqsourah)
      {
        replace char by vowel /a/;
      }
    // Ligature
    else if (the char is a Ligature)
      {
        insert ligature phonemic sequence;
      }
     // Tanween
    else if (the char is a Tanween character)
      {
        insert appropriate phonemic sequence for Tanween;
      }
     // Gemination
    else if (the char is a gemination sign)
      {
       repeat and insert the char before the gemination sign;
      }
  }
Continue;
```

Fig. 1. Pseudo-code for grapheme-to-phoneme conversion.

```
// Grapheme_To_Phoneme  algorithm (continuation)

 //Scan the grapheme buffer left-to-right
While (there is a word in the grapheme buffer);
  {
    Set output phonemic word buffer to NULL;
    Read the next graphemic word from the grapheme buffer;

    // Lookup the special words lexicon
    If (graphemic word is in the special spelling lexicon)
       {
         Read the word spelling from the lexicon;
         Concatenate the output phonemic word by the spelling;
       }

     // Apply the remaining ordered rules (rules 8 to 11, Shamsi & Ghamari, long vowel,
     // diphthong and short vowel generation)
     else if (graphemic word is not in the special spelling lexicon)
       {
        Position focus window at start of the  graphemic word;
        Set focus window size to M (M is the largest size in the rules set);
        // Scan the current graphemic word left-to-right by M, M-1, until 1
         While (M is not less than 1)
             {
               Position focus window to beginning of the graphemic word;
               Read M characters from the graphemic word;
                While (not end of the graphemic word)
                   {
                     //Scan the ordered rules top-to-bottom in the way in which they are
                        ordered, large-size rules first
                     While (not end of the ordered rules)
                        {
                          If (there is a rule matching the M characters in the focus window)
                             {
                               Read the phonemes corresponding to the rule;
                               Concatenate the output phonemic word by the phonemes matching
                               The rule;
                               Skip to right in the graphemic word by reading M characters;
                             }
                        }
                   }
                  Decrement M by 1;
        }  // End of else if (..)
  }

CALL Syllabification_Diphthong_Generation algorithm;
```

Fig. 1. (*continued*)

Pharyngealization of Arabic sounds is both hetrosyllable and tautosyllabic phenomenon. It is both a forward and a backward phenomenon. It affects all the six Arabic vowels (/a/, /i/, /u/, /a:/, /i:/ and /u:/), the two diphthongs (/aw/ and /aj/), the counterparts (/t/, /d/, /s/ and /D/) of the Arabic emphatics (/t'/, /d'/, /s'/ and /D'/), the sonorant /l/ and the trill /r/. The vowels, the diphthongs, the sonorant /l/ and the trill /r/ become heavily pharyngealized whenever they occur in a context that make them pharyngealized. The emphatic counterparts are assimilated to the sounds of the emphatics themselves. Examples of vowel and diphthong pharyngealization and emphatic assimilation in SA are found in the following words:

(1) “سطع” [Sa’Ta’Ea] (shines). Both the first and second short vowels, /a/ are pharyngealized and pronounced as [a’]. Since the first short vowel /a/ is pharyngealized, the phoneme /s/ is assimilated to the emphatic counterpart /s’/.

(2) “سيطر” [s’aj’t’a’r] (to control). The diphthong /aj/ is pharyngealized and pronounced as [aj’]. The second short vowel /a/ is also pharyngealized and is pronounced as [a].

The Arabic vowels and diphthongs are nasalized whenever they are followed by a nasal sound (/m/ or /n/). Anticipatory coarticulations like sound overlapping and assimilation are encountered with the Arabic voiceless stops /t/ and /k/ when followed by back or front high long vowels /u:/ or /i:/. In the context of speech synthesis, the phonetic variations of Arabic were discussed by (El-Imam, 1990).

Three categories that include twenty-two phoneme-to-phone rules have been defined. The rules cover the important phonetic variants of the basic phonemes described above. The phonetic variations’ rules are:

Fourteen rules to cover the pharyngealization of the six vowels, the two diphthongs, the sonorant /l/, the trill /r/ and the assimilation of the emphatic counterparts. These sounds undergo their respective phonetic changes whenever they are in the same syllable or in the neighborhood of an emphatic or another heavily pharyngealized sound. The rules are of the format;

[V] → [V’]/X_Y X,Y (an immediately neighboring syllable in which any of the emphatics (“ط”, “ظ”, “ص” “ض” or another heavily pharyngealized sound is present or just an immediately neighboring pharyngaelized sound)). V is any of the six vowel phonemes (/a/, /u/, /i/, /a:/, /u:/ and /i:/) and V’ is the pharyngealized counterpart of V. We observe the need for syllabification before the pharyngealization rules can be applied.

For the diphthongs, the rules are of the form,

$$[\text{aj or aw}] \rightarrow [\text{aj' or aw'}]/\text{X\_Y}, \quad \text{X}, \text{Y} = \{\text{same context given with the vowels above}\},$$

aj or aw are non-pharyngealized diphthongs and the aj’ or aw’ are their pharyngealized counterparts. An example of pharyngealization of the diphthong /aj/ is already given. Pharyngealization of the dipthong /aw/ is seen in the word: “طورَ” [t′aw’war] (to develop). The dipthong /aw/ is pharyngealized and pronounced as [aw’] because of the presence of the emphatic /t′/ in the same syllable with it.

For the assimilation of the emphatic counterparts and the pharyngealization of the sonorant /l/ or the trill/r/ the rules are of the form,

$$[\text{NE or C}] \rightarrow [\text{E or C'}]/\text{X\_Y}, \quad \text{X}, \text{Y} = \{\text{same context given with the vowels above}\}.$$

NE is any emphatic counterpart and E is the corresponding emphatic. C is either the sonorant /l/ or the trill /r/ and C’ is its pharyngealized counterpart.

Six rules to take care of the nasalization of any of the six vowels whenever any of these sounds is followed by a nasal sound. The rules are of the form,

$$[\text{V}] \rightarrow [\text{Vn}]/\text{X\_Y}, \quad \text{X} = \{\text{any grapheme that represents a consonant}\}, \ \text{Y} = \{\text{m or n}\}.$$

V is any of the six vowel phonemes (/a/, /u/, /i/, /a:/, /u:/ and /i:/). Vn is the nasalized counterpart of V.

Two rules to take care of the overlapping of the voiceless stop /t/ and the adaptation of the voiceless stop /k/ when followed by the long vowels /u:/ or /i:/. The rules are of the form,

$$[\mathrm{C}] \rightarrow [\mathrm{Co\ or\ Ca}]/\mathrm{X\_Y},$$
$$\mathrm{X} = \{\text{any grapheme that represents a consonant}\},\quad \mathrm{Y} = \{\text{u: or /i:/}\}.$$

C is either /t/ or /k/ and Co or Ca is either overlapped /t/ or adapted /k/.

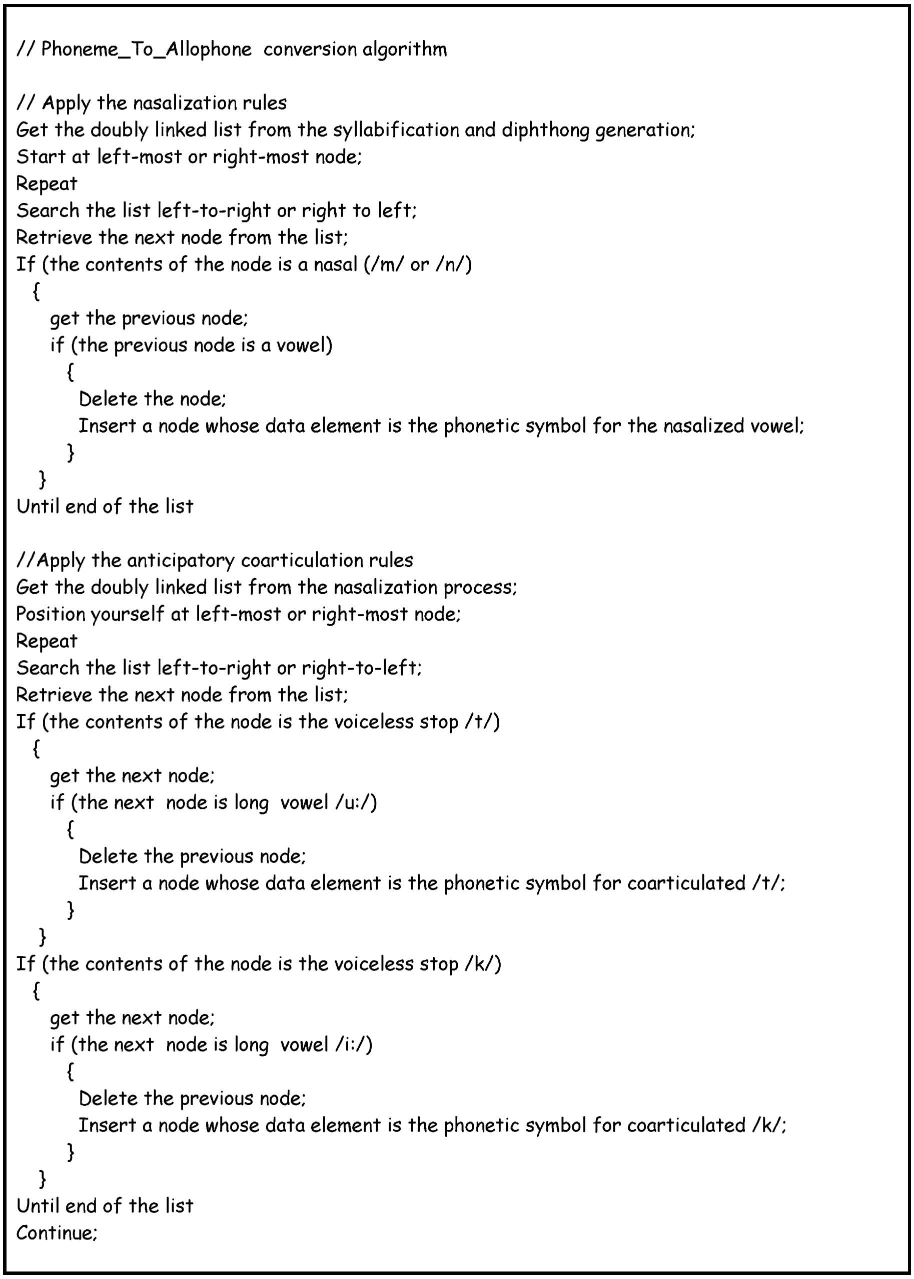
Pharyngealization is a dominant phonetic property of Arabic speech, i.e., if pharyngealization and another phonetic phenomenon affect a sound at the same time, pharyngealization will dominate. For example, in the word “صنع” /s’ana?’/ (to manufacture), the first vowel /a/ is both pharyngealized and nasalized. Because of the strong influence of pharyngealization, the vowel is considered as pharyngealized. For this reason the phonetic rules are also ordered so that pharyngealization rules are executed last. This and other considerations have implications on the type of data structure used to represent the phonetic string. The algorithm for realizing the phonetic rules is presented by the pseudo-code shown in Fig. 2.

5. Realization of the phonemic and phonetic rules

The pharyngealization of certain Arabic sounds and diphthong generation rules require that the phonemic string be organized into syllables. Syllabification is the first step, which is performed on the phonemic sequence. As was suggested in section 2.6, the phonemic sequence can be divided into syllables by noting that, in Arabic, the nucleus of every syllable is a vowel. The following algorithm applies to syllable generation. The phonemic representation is scanned starting from its right end (right-to-left scanning) looking for vowels. Whenever a vowel is located, look for a syllabic match for any of the six types of Arabic syllables (as pointed out in Section 3.6, Arabic syllables are of type CV, CVC, CV:, CV:C, CVCC and the rare CV:CC, where C represent a consonant and a V a vowel). Whenever a syllable match is found, tags are inserted to mark the syllable borders. Repeat the above steps until all vowels in the phonemic sequence are consumed. The syllabification algorithm is presented in the pseudo-code shown in Fig. 3. The same pseudo-code shows how dipthongs are generated from the syllables (if they are not part of the phonemic ordered rule set).

As an example of the syllabification, consider the CV.CV.CV.CVC word “كتبهم” (their books) whose phonemic sequence is /kutubuhum/. Application of the algorithm will result in the following sequence of events: the vowel near the end is surrounded by two consonants, therefore a CVC syllable is recovered and the remaining consonant-vowel sequence is CV.CV.CV. The CVCVCV sequence ends with a vowel and has a consonant to its left, therefore a type CV syllable is recovered. The remaining CVCV sequence has a vowel at its end and a consonant on the left. Again a type CV syllable is recovered. The remaining CV sequence is a lone syllable of type CV.

After the segmentation, pre-processing and testing of the exceptional words or words of irregular spellings, the Elison and Sukoon deletion rules are applied first. These rules actually generate null sounds and can be considered as pre-processing for the grapheme-to-phoneme generation prior to the application of the ordered set of rules. Elision and Sukoon deletion rules are then followed by the other unordered rules Ya Maqsourah rule, the Glottal stop insertion rule, the Tanween rules, the gemination sign rule and the Arabic ligature rules. The remaining phonemic rules are ordered and applied in the following sequence:

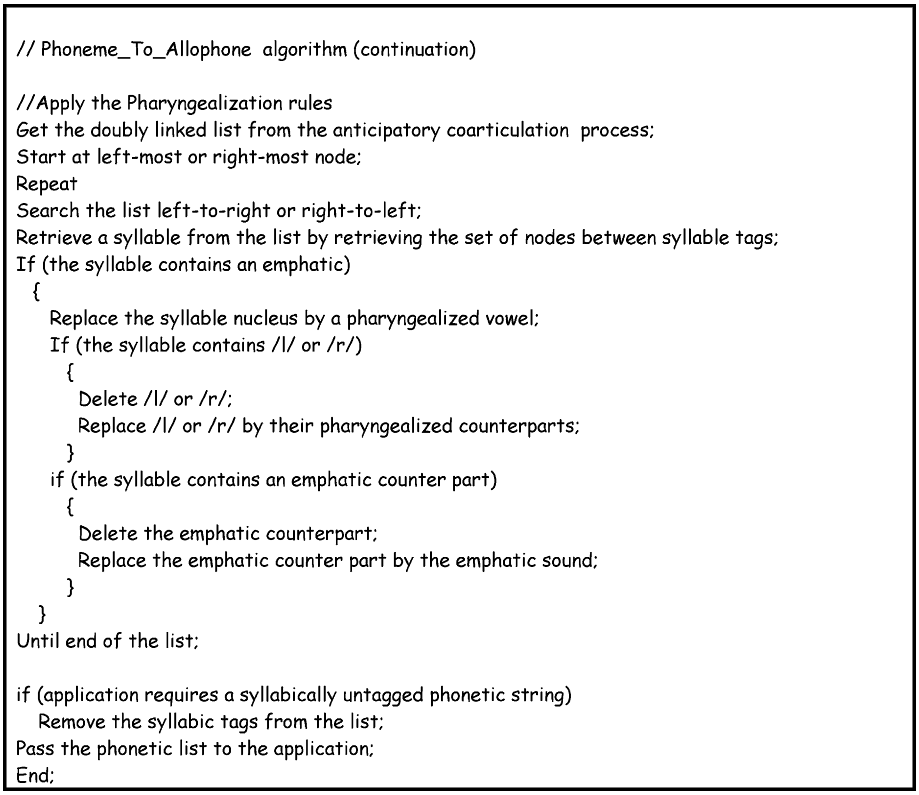


```
// Phoneme_To_Allophone  conversion algorithm

// Apply the nasalization rules
Get the doubly linked list from the syllabification and diphthong generation;
Start at left-most or right-most node;
Repeat
Search the list left-to-right or right to left;
Retrieve the next node from the list;
If (the contents of the node is a nasal (/m/ or /n/)
  {
    get the previous node;
    if (the previous node is a vowel)
      {
        Delete the node;
        Insert a node whose data element is the phonetic symbol for the nasalized vowel;
      }
   }
Until end of the list

//Apply the anticipatory coarticulation rules
Get the doubly linked list from the nasalization process;
Position yourself at left-most or right-most node;
Repeat
Search the list left-to-right or right-to-left;
Retrieve the next node from the list;
If (the contents of the node is the voiceless stop /t/)
  {
    get the next node;
    if (the next  node is long  vowel /u:/)
      {
        Delete the previous node;
        Insert a node whose data element is the phonetic symbol for coarticulated /t/;
      }
   }
If (the contents of the node is the voiceless stop /k/)
  {
    get the next node;
    if (the next  node is long  vowel /i:/)
      {
        Delete the previous node;
        Insert a node whose data element is the phonetic symbol for coarticulated /k/;
      }
   }
Until end of the list
Continue;
```

Fig. 2. Pseudo-code for phoneme-to-phone conversion.



```
// Phoneme_To_Allophone  algorithm (continuation)

//Apply the Pharyngealization rules
Get the doubly linked list from the anticipatory coarticulation  process;
Start at left-most or right-most node;
Repeat
Search the list left-to-right or right-to-left;
Retrieve a syllable from the list by retrieving the set of nodes between syllable tags;
If (the syllable contains an emphatic)
  {
    Replace the syllable nucleus by a pharyngealized vowel;
    If (the syllable contains /l/ or /r/)
      {
        Delete /l/ or /r/;
        Replace /l/ or /r/ by their pharyngealized counterparts;
      }
    if (the syllable contains an emphatic counter part)
      {
        Delete the emphatic counterpart;
        Replace the emphatic counter part by the emphatic sound;
      }
  }
Until end of the list;

if (application requires a syllabically untagged phonetic string)
   Remove the syllabic tags from the list;
Pass the phonetic list to the application;
End;
```

Fig. 2. (*continued*)

- The Shamsi and Ghamari rules and the ligature rules are applied next.
- The long vowels are generated according to the context of the short vowels.
- The remaining short vowel graphemes are replaced by their respective pronunciations.
- The diphthong rules and the grapheme-to-phone rules are applied (after syllabification).

In implementing the algorithm for applying the rules, the input grapheme words are kept in a buffer. The words are read sequentially from the buffer and each word is scanned left-to-right using the set of rules. Moving context and focus window (a moving window that focuses on the grapheme under investigation including its right and left context) is used on the word currently undergoing processing. The window size is initially set to the number of graphemes in the largest rule, M. The window is initially positioned at the beginning of the current word. M graphemes are read from the word and the ordered rules are scanned looking for a match. If a match is found, the corresponding phonemes are concatenated into a phoneme-output string and the focus window is shifted by M graphemes to the right. The scanning is continued from this position. If no match is found, the focus window size is decremented by one, the window is moved to the beginning of the word and M–1 graphemes are read from the word and the rules are scanned again until a match is found. Note that a match is bound to be found since single graphemes are associated with phonemes and always have pronounciations. When the focus window size becomes one, the end of the current word is reached and all rules are exhausted the window size is reset to

```
// Syllabification_Diphthong_Generation algorithm

// Syllabification
Start at the right-most node of the NULL terminated grapheme string;
Repeat
Retrieve  next node to the left;
If (Node contains a vowel)
   {
      Search left and right of the node;
       Form syllables starting from the longest syllabic pattern (CVCC) down to the
       Shortest (CV);
      Look for a match of any of the six syllable types;
      Skip one node to the left of the vowel and insert a node with a unique data type as
       a syllable Tag;
   }
Until one node position before left-most end of the list;

// Diphthong generation
Start at the left-most or the right-most node of the syllabically tagged  phonemic list;
Repeat
Search the list (right-to-left or left-to-right) looking for syllables of type CVC or CV:C;
If (syllable of type CVC or CV:C)
  {
    Check the syllable closing consonant;
    If (closing consonant is /w/ or /j/)
       {
         Delete node holding the closing consonant from the list;
         Insert a node containing the appropriate diphthong in the list;
       }
   }
Until end of the list is reached;

CALL Phoneme_To_Allophone algorithm;
```

Fig. 3. Pseudo-code for syllabification and diphthong generation.

its maximum value of M graphemes, another word is read from the buffer and the process is repeated.

6. Assessment of the Arabic letter-to-sound transcription system

There is an abundance of literature on the evaluation of the quality of synthesized speech produced by TTS systems (Silverman et al., 1990; Van Santen, 1993; Pols and Jekosch, 1997). However, there seems to be a lack of literature on objective evaluation of the performance of any of the individual TTS components such as the grapheme-to-phoneme conversion, which is an essential submodule of the letter-to-sound system. Grapheme-to-phoneme conversion finds application in areas other than speech synthesis. It is therefore essential to objectively evaluate its performance independent of or with little reference to any application.

There also seems to be a lack of a standard methodology to evaluate the NLP component of a TTS system. Different methods have been used, in the past, to evaluate the rule-based pronunciation component. Elovitz (Elovitz et al., 1976) based the evaluation of the performance of his rule set on frequency weighting and expected to correctly pronounce up to 90% of the words in a random sample of English text. Bernstein and Nessly (Bernstein and Nessly, 1981) used subsets of 1000 words from the Brown corpus, (Kucera and Francis, 1967) and achieved scores ranging from 65% (rarest subset) to 86.8% (most common subset) words correctly pronounced. Hunnicut (Hunnicutt, 1980) demonstrated preliminary evaluation of her rule set using subsets of 200 words from the Brown corpus and achieved scores ranging from 66% (rarest subset) to 100% (most common subset) words correctly pronounced. Using these small chunks of test data, Hunnicut used frequency weighing to estimate the performance of her rule set on words absent from the dictionary at 71% correct pronunciation. (Divay and Vitale, 1997) tested their newer rule set for English on word subsets of 19,837 words taken from the Brown corpus and achieved 64.37% words correct. Given all these non-standard evaluation methods, how would the performance of a rule-based pronunciation component of a TTS system be evaluated? Damper (Damper et al., 1999) suggested a standard procedure to be followed for the evaluation of the phonemization component: (1) automatic phonemization methods should be tested on the same large dictionaries in their entirety as this demonstrates clearly the asymptotic performance of the rule-based transcription on a large test data. (2) The use of a common standard metric, like scoring in terms of words correct, as this is more stringent and sensitive metric than phoneme correct metric since the words are either correct or not and (3) the use of a common list of symbols or a standard output phoneme set.

In reality, the capabilities of most TTS systems go far beyond the pronounciation of isolated words. Most TTS systems, including the Arabic TTS, are unlimited vocabulary dealing with pronounciation of words, phrases, sentences, abbreviations, proper names, symbols and in general any kind of written text. Given such variable input to the automatic phonemization component of a TTS system, it seems that restricting the test data to isolated words from a language corpus will not be general enough to test all aspects of the system. There are, also, numerous pronunciation ambiguities and the pronounciation of certain words may change according to their positions within the sentences. In Arabic, for example we saw, in Section 3.2, the compounding of words when a Shamsi rule applies immediately after a word that ends in a vowel. For such reasons an evaluation method for the grapheme-to-phoneme conversion will require more than isolated words and we need an evaluation corpus rich in phonemic and phonetic contents to test all the rules and the exceptions. Realizing such demands on the automatic phonetization component, Yvon and De Mareuil (Yvon et al., 1998; De Mareuil et al., 1998) used running text rather than lexica in a corpus of about 26,000 words organized in 2000 test sentences for testing eight French TTS systems. Damper (Damper et al., 1999) used 16,280 words from the American English Teachers' Word Book (TWB) of (Thorndike and Lorge, 1944) to test Elovitz (Elovitz et al., 1976) rules in comparison to three other data-driven methods including PbA.

Arabic is certainly not as complex as French or English and it does not have the same spelling irregularities as those languages have, which require extensive rule sets and huge special lexicons. Because of the regularities of Arabic spelling, it is possible to device a rule set which is generic enough to cover all the spelling rules of the language. The exceptions (which predominantly fall under the category of violations to the generation of long vowel /a:/) to the regular spelling rules could then be handled by a dictionary of exceptional words which has to be parsed prior to executing the generic rules. The only issues that we should pay special attention to is that the test data should be comprehensive and arbitrary enough to include every rule, every exceptional case and every phoneme/phone to be generated as many times as possible. We could therefore limit the evaluation corpus for the Arabic letter-to-sound conversion to an assortment of words and sentences rich enough in phonemic and phonetic contents, which will test the major difficulties in Arabic letter-to-sound transcription. These complexities have already been discussed in this article and they include complicated phonetic structures (complicated grapheme-to-phoneme rules and phoneme-to-phone rules), acronyms, abbreviations and symbols, numerals, proper names, segmentation, etc. Essentially the corpus should be rich in phonetic content and should include all the pronounciation variants of Arabic speech.

One method to evaluate the performance of grapheme-to-phoneme conversion would be to compare its output phonemic string with a reference transcription and then count any discrepancies as errors. Such an approach would concentrate on the transcription of isolated words taken from phonetic dictionaries.

6.1. The Arabic evaluation corpus

The Arabic letter-to-sound transcription system was tested on a corpus of 6000 most frequently used Arabic words, which were compiled by a team from the University of Umm Alqura (University of Umm Alqura, Institute of Arabic Studies, 1970). And a list of 4000 popular Arabic proper names compiled by (Humoud, 1995) as well as acronyms, abbreviations and symbols and numbers, which are externally supplied. Rule-based methods are not like trainable data-driven approaches in that the data should be divided into training and test data, which are independent of each other. A rule is general and when it is applicable, it applies irrespective of the class of word. However, if one could classify training words as those used as examples of words that bear the rules and test words as those used to test the performance of the rules, we could say that the choice of the test words is arbitrary. Table 4 shows the occurrences of the phonemic and phonetic rules in some of the test words. The statistics shown in Table 4 demonstrate that every rule that matches the spelling system of Arabic is covered by an adequate number of test words from Umm Alqura list. The Umm Alqura list also contains numerous Arabic words that do not obey the regular spelling rules (the special words or words of irregular spellings). The list does not include any acronyms, abbreviations and symbols or numbers so these are externally acquired and added to the test words.

If we define rare words as words outside the most frequent used vocabulary (outside the Umm Alqura list). As far as the relationship between the orthographic form of SA words and their sound contents, rare and most frequently used words share the common property that they obey the letter-to-sound rules, as long as they are not exceptional words or names as discussed in Section 4.1. This is true for SA as any Arabic word is subjected to the same generic spelling and pronunciation rules as long as its not an exception. Of course, there are exceptional words that are rarely used and others that are used most frequently, which imply that the designer of the dictionary of exceptions has to make sure that his list of exceptions is complete. This is what happens in this system. The list of exceptional words and names are derived from two of the most comprehensive lexicons for SA (Al-Fairuz Abadi, 1996; Humoud, 1995).

Table 4
The occurrences of the phonemic and phonetic rules in the some of the test words

Rule type	Approximate number of occurrences
Phonemic rules	
Sikoon deletion	1845
Elision of Alef[a]	500
Ya Maqsourah	95
Glottal stop insertion	1260
Tanween[b]	3500
Gemination	2100
Ligatures	1100
Shamsi and Ghamari	3000
Vowelization or Mad	3500
Diphthongs	600
Short vowels (Fathah, Damah and Kasrah)	12,000
Phonetic rules	
Pharyngealization of vowels	600
Nazalization of vowels	1087
Emphasis of /l/ and /r/	60
Sound assimilation due to emphasis	74
Overlapping and adaptation	30

[a] Obtained by converting around 500 past form of verbs to plural forms and includes the elision of Hamzat Wasl.

[b] Most words in Umm Alqura list ends with Tanween Damah. The endings are changed to reflect Tanween Fathah or Kasrah.

The corpus is organized into test items made of isolated words (including regular words, proper names, acronyms, abbreviations and symbols and numbers) and test sentences. The test items were phonetized by the grapheme-to-phoneme and the phoneme-to-phone components of the letter-to-sound system. The test items are also manually transcribed to provide the reference transcription.

Regarding the criteria put forward by (Damper et al., 1999), we have used the two scoring metrics i.e., a word correct (stringent metric) and a phoneme or symbol correct metrics. The phonemic symbols used are the standard SAMPA for Arabic. For obvious reasons, the phoneme correct score should be higher than the more stringent word correct score since an error in just one phoneme/phone could affect many words. Having the previous points in mind, the scoring scheme adopted in evaluating the system was to count as an error any discrepancy between the reference transcription and the transcription produced by the Arabic phonetizer. The corpus (Umm Alqura list, the list of proper Arabic names and the externally acquired material) contains an approximate total of 50,000 phonemic and phonetic symbols, which are distributed among all the phonemic and phonetic transcription rules. Most of the words in the Umm Alqura list are commonly used and have a good coverage of all the Arabic letter-to-sound rules presented in this article. The list of popular Arabic proper names contains Arabic male and female names. The names also have a good coverage of the Arabic letter-to-sound rules.

Arabic grapheme-to-phoneme rules can be ordered from the most difficult to the least difficult in the following manner:

(1) The Shamsi rules and the merging of two words when a Shamsi word is preceded by a word whose last consonant is vocalized.
(2) The Diphthong generation rules.
(3) The rest of the grapheme-to-phoneme rules: The Ghamari rules. The long vowel generation rules. The short vowel replacement rules. The Tanween rules. Arabic ligature rules. The deletion of "Alef" when it occurs as Hamzat Wasl. The repetition of the grapheme when Shedda appears on it. The Ya maqsourah rule. The Sukoon deletion rules. The one-to-one grapheme-to-phoneme rules. All these rules require approximately the scanning of either the present character or one character ahead of it and are relatively easy to handle.

The Umm Alqura test list of frequently used words has about 30% (around 2000 Shamsi words) of the total words in the list in which the Shamsi characters appear at the beginning of the words. Around 80% of these 2000 Shamsi words are nouns or adjectives and hence they are liable to having the definite article Alef Lam "ال" being appended to their beginnings to fulfil the Shamsi transcription rule. The 2000 Shamsi words constitute a reasonable vocabulary of test items to test the Shamsi transcription rule.

To enhance the assessment, around 100 sentences were formulated to cover the merging or compounding of the previous word with the current Shamsi word. The 100 sentences span all the 14 Shamasi characters and all the previous vocalic signs at the end of the previous word. For example, in the phrase "طلعت الشَمس" which, is pronounced (after application of the grapheme-to-phoneme and phoneme-to-phone rules) as [t'a'l'?'ati?aSamsu], the first word "طلعت" is merged with the Shamsi word "الشَمس" after the grapheme-to-phoneme transcription of the two words.

The Umm Alqura test list of words contains around 10% words (about 600 words) that bear dipthongs. This is also a good number of words to provide a test bed for diphthongs. The other rules are also well represented in Umm Alqura list. Examples of words used in the assessment lists are shown in Table 5.

Other lists of test items were derived from the Umm Alqura list and the list of Arabic proper names and are used to test the phoneme-to-phone component of the system. Again the rule-set here is arranged in descending order of difficulty from the pharyngealization rules, followed by nasalization rules and down to the overlapping and adaptation rules. Representative words from these lists are shown in Table 5. The popularity and rarity of some of the Arabic rules is demonstrated in Table 4, but above all this table shows that the least occurring rule (overlapping and adaptation in the phonetic rules) has been tested 30 times. While the most occurring rule (the short vowels generation) have been tested more than 12,000 times.

The test is automated by creating an electronic dictionary, which includes for each test entry (grapheme string) its corresponding reference phonemic/phonetic string. The test items were presented to the phonetizer by preparing text files from the test items and the files are used as input to the letter-to-sound transcription system. The output of the phonetizer is pure phonemic string if the test item obeys only the grapheme-to-phoneme rules and phonetic when the test item obeys both grapheme-to-phoneme and phoneme-to-phone rules. The output of the phonetizer is compared to the reference phonemic/phonetic string. For every test item, any discrepancy between phonetizer output and the corresponding reference transcription is counted as an error. A transcription error of one phoneme causes a test-word error or a group of test-words errors (word incorrect errors) and these errors are accumulated in counter(s) of word errors that cumulatively counts the number of test-words, which are in error. Likewise different error counters are

Table 5
Example words categorized by rule type and used in assessing the transcription

Rule type	Total number of test items, words/ sentences tested	Example words/sentences
Shamsi rule	Approximately 2000 words or test items	“التِمساح” /?attimsa:X\/ (the crocodile); “الثمر” /?aTTamar/ (the fruit); “الدِفاع” /?addifa:?'/ (the act of protecting); “الذَيل” /?aDDajl/ (the tail); “الرُمَان” /?arrumma:n/ (a kind of fruit); “الزائر” /?azza:?ir/ (the visitor); “السَاحل” /?assa:X\il/ (the beach); “الشَارع” /?aSSa:riE/ (the road); “الصُحف” /?as's'uX\uf/ (the news papers); “الضَجيج” /?ad'd'agi:g/ (the noise); “الطُرق” /?at't'uruq/ (the roads); “الظِلال” /?aD'D'ila:l/ (the shadows); “اللُغه” /?alluGah/ (the language); “النُجوم” /?annugu:m/ (the stars)
Merging of a previous word whose end is vocalized with the next Shamsi word	Approximately 100 sentences	“حضرت الليله” /X\ad'artu?allajlah/ (I came tonight); “جاء الطُلاب” /ga:?a?at't'ula:bu/ (the students came); “جئت الليله؟” /gi?ti?allajlah/ (did you come tonight?)
Diphthong generation rules	Approximately 600 words	“إستورد” /?istawrad/ (he imported); “إستيقظ” /?istajqaD'/ (he woke up); “إتفاقيَه” /?itifa:qiIIah/ (agreement); “أخوَه” /?uxuwwah/ (brotherhood); “أوراق” /?awra:q/ (papers);
All other rule types	Approximately 8000 regular words and common Arabic proper names	“ربَى” /rabba/ (to bring up a child); “إستئناف” /?isti?na:f/ (to make an appeal); “ذهبوا” /Dahabu:/ (they went to); “جميلا” /gami:lan/ (beatiful); “فأنظر” /fanD'ur/ (look at); “جمال” /gama:l/ (beauty); “راع” /ra:?'in/ (care taker); “راكب” /ra:kibun/ (passenger); “رتَب” /rattaba/ (he arranged); “لأنَ” /la?anna/ (because of); “لام” /la:ma/ (to blame); “الآن” /?ala:n/ (now); “كتب” /kutiba/ (it was written); “سونيا” /sɑanjΓ/ (Sonia, in English)
Special words, abbreviations, symbols and acronyms	More than 300 test items	“هذا” /ha:Da:/; “هؤلاء” /ha:?ula:/; “ممَ” /mima:/; “طه” /t'a:hah/; “الرحمن” /?arraX\ma:n/; “اليونسكو” (UNESCO) /junesk ∂v/ (United Nations Educational, Scientific and Cultural organization)
Pharengealization	More than a 1000 words	“رابطه” [ra:bit'a'h] (union); “ضبَاط” [d'u'bba:'t'} (officers); “صدق” [s'i'd'kun]; “ضابط” [d'a:'bi't'un] (an officer); “طول” [t'u:'l'u'n] (being long); “طين” [t'i:'nun] (clay); “طيب” [Taj'hibun] (pleasant); “طور” [t'uw'wir] (developed); “صلى” [s'a'l'l'a'] (he prayed); “صرف” [s'a'r'fun] (giving money)
Nasalization	More than 500 words	“دمعه” /da^{n} m?'ah/ (tear); “دمر” /du^{n} mmara/ (destroyed); “خادم” /xa:di^{n} mun/ (servant); “خام” /xa:^{n} mun/ (raw material); “دون” /du:^{n} na/ (below the standard); “ديمقراطيَه” /di:^{n} maqra:tijjah/ (democracy)
Adaptation and overlapping	More than 500 words	“قوت” /qu:tun/ (living); “وكيل” /waki:lun/ (agent)

maintained by the system for each phoneme/phone. Each error counter (word or phoneme/phone) is automatically updated when an error occurs.

If the goal of phonetization is to develop a TTS system, the scoring strategy of counting every discrepancy from the reference transcription as an error is an oversimplification since not all kinds of transcription errors equally impair the intelligibility of the output speech. For example, some errors resulting in overlooking the allophonic variant of a phoneme have little or no bearing on the ineligibility of the output speech. Nevertheless, this scoring approach is used in the present system since we are evaluating the grapheme-to-sound component as an independent component of the TTS system.

6.2. Evaluating the test results and analysis of transcription errors

In general, the letter-to-sound system reported few transcription errors on both words and phonemes/phones. The overall score of the system is over 98% phonemes correct while the percentage of correctly pronounced words is around 92% correct words. The sources of the errors are irregular words, abbreviations and symbols missing from the lexicon of special words or some proper names producing wrong pronunciations or words in the test files are misspelled. The phonemic/phonetic output is a function of the precision of the letter-to-sound rules and the completeness of the exceptional dictionary. In this system, we have included every Arabic spelling rule we have encountered in the literature of Arabic spelling books reported in this article and every Arabic word with irregular spelling. However, to analyze the errors, a list of erroneous words, which are in error was compiled. The errors were classified according to the following:

(1) The grammatical category (proper name, abbreviation or symbol, number or any other lexical item) of the erroneous word that caused the error.
(2) The error type or the submodule of the letter-to-sound system that caused the error (the submodules of the Arabic letter-to-sound system are the segmentation, the pre-processing, the grapheme-to-phoneme conversion and the phoneme-to-phone conversion).

The distribution of errors by word category is shown in Table 6. This table illustrates that a good percentage of errors are encountered with some proper names and acronyms (the percentage of errors in these categories are 19.2% and 18% or around 37.2%). Some of the Arabic proper names are of foreign origins and have pronounciations that are different from the SA standards. For example, the name ‘‘سونيا’’ (Sonia, in English) is perhaps of European origins but is used in Arabic and pronounced by most people correctly according to its English pronounciation /sɑnjΓ/. The present system transcribes it as /su:nja:/, which is obviously erroneous when, compared to the English transcription. Another example, is the acronym ‘‘اليونسكو’’ (UNESCO) (United Nations Educational, Scientific and Cultural Organization) which is transcribed by the Arabic phonetizer as /?alju:nisku:/ while its proper English transcription would be /junesk ∂ ʋ/. These types of errors fall in the category of foreign names whose ultimate solution will be an English synthesizer

Table 6
Distribution of errors according to word category used for testing

Word category	Total number of errors	Total system errors (%)
Proper names	160	19.1
Acronyms	150	18
Numbers	120	14.5
Symbols	50	6
Abbreviations	40	4.8
All other word types	310	37.3
Total words in error	830	

Table 7
Distribution of errors according to letter-to-sound submodule or error type

Error type (submodule)	Total number of Errors	Total system errors (%)
Segmentation and pre-processing	360	43.4
Grapheme-to-phoneme conversion	290	35
Phoneme-to-phone conversion	180	21.7

imbedded with the Arabic one. While such test tokens constitute less than 2% of the total words in the test corpus, yet they contribute to most of the errors in the erroneous words.

Table 7 represents the distribution of the errors according to their types or the submodules that cause these errors. It appears that most causes of errors originate in pre-processing (acronym, abbreviations and symbols, and number conversion) and the lexicon look-up stages of the system either as test items, which are missing from the lexicon or test items, that produce erroneous pronounciation.

Despite these errors, the author is of the opinion that conversion of Arabic text to phonemes and phones can best be done by a complete set of letter-to-sound rules and a complete dictionary of irregular words, abbreviations and symbols and acronyms. To deal with foreign words and foreign proper names, the system needs to be enhanced with foreign language synthesizers. This would be predominately an English synthesizer with regards to its influences on the Arabic language.

7. Conclusions

This article presented an in-depth analysis of the problems of converting Arabic text into sounds, the composition of the transcription rules, the development of algorithms to implement the rules and the assessments of the output of this important natural language processing component of any TTS system. It is impossible to focus on these issues on articles dedicated to speech synthesis and yet give a fair treatment of the other facets of the diverse synthesis problem. Phonetization of text is essential for TTS conversion and it can find applications in speech recognition and in NLP for education.

In general there are problems associated with converting any writing system from its orthographic form to sequences of phonemes and phones. Such problems have various degrees of complexities depending on the writing system for which phonetization is sought. It all depends on the degree of correspondence between the writing system and the sound system of the language under investigation. Complex systems like English or French are characterized by lack of correspondence between the spellings and their phonetic realizations. Trivial systems like Swahili or Spanish have a high degree of correspondence between spelling used and its phonetic realization. Arabic is in between English/French and Swahili/Spanish. The Arabic orthographic system shares with English and French certain irregularities such as morphophonemic problems, elision, the presence of foreign words and names and other specific spelling irregularities. All these problems have been discussed in this article and solutions were suggested.

The article targeted four main issues related the Arabic text transcription or conversion to sound. They are: (1) it discussed and dealt with the problems related to converting Arabic orthography into phonetic sequences. (2) It presented to the reader a comprehensive set of letter-to-sound rules (phonemic and phonetic) for Arabic text transcription. (3) The rules were put in a framework, which is more suited for computer-based implementation, which is an essential component of any TTS system and (4) it presented an assessment of the performance of the Arabic letter-to-sound transcription.

There are three components to a letter-to-sound transcription system: segmentation and pre-processing component, a grapheme-to-phoneme component and a phoneme-to-phone component. For the grapheme-to-phoneme component, 11 categories of rules were developed. The specific rules within these categories handle the basic characteristics of the Arabic writing system, which include: (1) omission and insertion of certain graphemes. (2) Elision rules, (3) Shamsi and Ghamari rules. (4) Tanween rules. (5) Gemination rules. (6) Arabic ligature rules. (7) Long vowel generation rules. (8) Diphthong generation rules and (9) short vowel generation rules. There are certain Arabic words, which do not obey the regular spelling rules. In Arabic there are also abbreviations, symbols, acronyms, numbers and words with irregular spellings. To handle these exceptions, a dictionary of exceptions that include the words together with their correct pronounciations were defined and created.

An essential component of the letter-to-sound transcription system deals with the generation of the actual sounds or phones of the language from the phonemic sequences. An important tool for this part of the system is the generation of the Arabic syllables from the phonemic sequences. This is essential since some of the important phonetic variations of Arabic, like pharyngealization, are best dealt with if the syllables are known. Generating the syllables is also important because they can be used to ease the application of certain letter-to-sound rules and the syllables can also be used to carry information about intonation and prosody of Arabic speech in any future Arabic TTS system, which envisages a naturally sounding speech. The syllables of Arabic were presented and generated using the property of Arabic that the nucleus of every Arabic syllable is a vowel. For the phoneme-to-phone component of the system, 22 rules were defined. The rules take care of the important phonetic variations of the language such as pharyngealization, sound assimilation, nasalization, overlapping and adaptation.

When any problem, such as the problem of converting Arabic text to sounds, is to be solved and realized as a computer application, that problem must be formulated in algorithmic form suitable for coding in a computer language that the machine understands. In this respect, the letter-to-sound rules were ordered in such a way that they do not contradict each other. Suitable algorithms have been developed and were put in the form of a pseudo-code, which an interested user can transform, with moderate effort, into a computer code using an appropriate computer language of his choice.

The performance letter-to-sound rules were tested using a list of most frequently used Arabic words and proper names. The results of the tests showed that the accuracy of the present system is very high. The accuracy of letter-to-sound conversion is a function of the pre-processing, the precision of the letter-to-sound rules and the completeness of the lexicon of words with irregular spelling, abbreviations and symbols and acronyms.

References

Ainsworth, W.A., 1973. A system for converting English text into speech. IEEE Transactions on Audio and Electroacoustics AU-21, 288–290.

Al-Ani, S.H., 1970. Arabic Phonology. Mouton, The Hague.

Al-Fairuz Abadi, M.M., 1996. Al Qamous Almuhiet. Arresalah Publishers, Beirut.

Allen, J., Hunnicutt, M.S., Klatt, D., 1987. From Text to Speech – The MITalk System. MIT Press, Cambridge, MA.

Anis, I., 1992. Language Phonetics (in Arabic). The Anglo-Egyptian Book Publishers, Cairo.

Bagshaw, P.C., 1998. Phonemic transcription by analogy in text-to-speech synthesis: novel word pronunciation and lexicon compression. Computer Speech and Language 12 (2), 119–142.

Balabaki, R., 1981. Arabic and Semitic Writing Systems: Studies into the Origins and History of Semitic Writing. The Science for Millions, Bairot.

Beesley, K., 1996. Arabic Finite-State Morphological Analysis and Generation. COLING-96.

Beesley, K., 1998. Arabic Morphological Analysis on the Internet. In: Proceedings of the Sixth International Conference and Exhibition on Multi-lingual Computing, Cambridge, England.

Belhoula, K., 1993. Rule-based grapheme-to-phoneme conversion of names. Proceedings of the Eurospeech, 881–884.

Belrhali, R.V., Auberge, V., Boe, L.J., 1992. From lexicon to rules: towards a descriptive of french text-to-phonetics transcription. In: Proceedings of the International Conference on Spoken Language Processing 92, Alberta, pp. 1183–1186.

Bernstein, J., Nessly, L., 1981. Performance comparison of component algorithms for the phonemization of orthography. In: Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics, Stanford, CA, pp. 19–21.

Chomsky, N., Halle, M., 1968. The Sound Pattern of English. Harper and Row, New York.

Coker, C.H., 1985. A Dictionary-Intensive Letter-to-Sound Program. JASA 78 (Suppl. 1), S7.

Coker, C., Church, K., Liberman, M., 1990. Morphology and rhyming: two powerful alternatives to letter-to-sound rules for speech synthesis. In: Bailly, Benoı̂t (Eds.), Proceedings of the First ESCA Workshop on Speech Synthesis, Autrans, European Speech Communication Association, France, 1990, pp. 83–86.

Daelemans, W., van den Bosch, A., Weijters, T., 1997. IGTree: using trees for compression and classification in lazy learning algorithms. Artificial Intelligence Review 11, 407–423.

Damper, R.I., Eastmond, J.F.G., 1997. Pronunciation by analogy: impact of implementational choices on performance. Language and Speech 40 (1), 1–23.

Damper, R.I., Marchand, Y., Adamson, M.J., Gustafson, K., 1999. Evaluating the pronunciation component of text-to-speech systems for English: a performance comparison of different approaches. Computer Speech and Language 13 (2), 155–176.

Damper, R.I. (Ed.), 2001. Data-Driven Methods in Speech Synthesis. Kluwer Academic Publishers, Dordrecht.

Dedina, M., Nusbaum, H., 1991. PRONOUNCE: a program for pronunciation by analogy. Computer Speech and Language 5, 55–64.

De Mareuil, P., Yvon, F., D'Alessandro, C., Auberge, V., et al., 1998. Evaluation of grapheme-to-phoneme conversion for text-to-speech synthesis in French. In: Proceedings of the First International Conference on Language Resources and Evaluation, Grenade, pp. 641–646.

Department of Phonetic and Linguistics, 2002. Speech Assessment Methods Phonetic Alphabet (SAMPA) for Arabic. Department of Phonetic and Linguistics, University College London (UCL), UK.

Divay, M., Vitale, A.J., 1997. Algorithms for grapheme–phoneme translation for English and French: applications for databsae searches and speech synthesis. Computational Linguistics 23 (4), 495–523.

Dutoit, T., 1997. An Introduction to Text-to-Speech Synthesis. Kluwer Academic Publishers, Dordrecht.

Dutoit, T., et al., 2000. Euler: an open, generic, multi-lingual and multi-platform text-to-speech system. In: Proceedings of LRECO'00, Athens, pp. 563–566.

El-Imam, Y.A., 1990. Speech synthesis using partial syllables. Computer Speech and Language 4, 203–229.

El-Imam, Y.A., 2001. Synthesis of Arabic from short sound clusters. Computer Speech and Language 15 (4), 355–380.

Elnaggar, A., 1992. A phrase structure grammar of the Arabic language. Journal of Natural Language Processing, Special Edition of Coling' 1990 189 (1).

Elnaggar, A., 1993. A finite state automata of the Arabic grammar. Journal of Natural Language Processing 19 (9).

Elovitz, H.S., Johnoson, R., McHugh, A., Shore, J.E., 1976. Letter-to-sound rules for automatic translation of English text to phonetics. IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-24, pp. 446–459.

Elshishini, H., Elnaggar, A., 1994. Two Arabic syntax analyzers, vol. 38, No. 3, IBM Scientific Center, Cairo.

Gibbon, D., Moore, R., Winski, R. (Eds.), 1997. Handbook of Standard and Resources for Spoken Language Systems. Mouton de Gruyter, Berlin.

Hassanain, A.T., Shahata, H., 1998. Rules for Arabic writing: Theory and Applications, Arabic Book House, Cairo (in Arabic).

Hertz, S.R., 1979. Appropriateness of different rule types in speech synthesis. In: Wolf, J.J., Klatt, D.H. (Eds.), Speech Communication Papers, No. 50, Acoustical Society of America, pp. 511–514.

Humoud, M., 1995. The Encyclopedia of Arabic Names and Their Meanings. Dar Al fikr Al libnani, Beirut.

Humoud, M., 1998. The Encyclopedia of Arabic Spelling and Grammar. Dar Al fikr Al libnani, Beirut.

Hunnicutt, S., 1980. Grapeme to phoneme rules: a review, STL-QPSR 2–3.

Kucera, H., Francis, W.N., 1967. Computational Analysis of Present-Day American English. Brown University Press, Providence, RI.

Laporte, E., 1988. Methodes et lexicales de phonetisation de textes, These, Univsite de Paris 7.

Levinson, S.E., Olive, J.P., Tschirgi, J.S., 1993. Speech synthesis in telecommunications. IEEE Communications Magazine, 46–53.

Liberman, M.Y., Church, K.W., 1992. Text analysis and word pronunciation in text-to-speech systems. In: Furui, S., Sohndi, M. (Eds.), Advances in Speech Signal Processing. Dekker, New York, pp. 791–831.

Luk, R., Damper, R., 1996. Stochastic phonographic transduction for English. Computer Speech and Language 10, 133–153.

Marchand, Y., Damper, R.I., 2000. A multi-strategy approach to improving pronunciation by analogy. Computational Linguistics 26 (2), 195–219.

Matsumuto, T., Yamaguchi, Y., 1990. A multi-language text-to-speech system using neural networks. In: Bailly, G., Benoit, C. (Eds.), Proceedings of the ESCA Workshop on Speech Synthesis. ESCA, Autrans, pp. 269–272.

McAllister, M., 1989. The problem of punctuation ambiguity in full automatic text-to-speech conversion. In: Proceedings of the Eurospeech 89, vol. 1, Paris, pp. 538–541.

McIlroy, M., 1974. Synthetic English Speech by Rule, Bell Telephone Laboratories Memo.

Meng, H.M., 1995. Phonological parsing for bi-directional letter-to-sound and sound-to-letter generation, Ph.D. thesis, MIT Press, Cambridge, MA.

Mitchell, T.F., 1952. Prominence and Syllabification in Arabic. Bulletin of School of Oriental and African Studies, vol. 23, London, pp. 269–289.

Qazzi, N.M., 2000. Summary of Arabic writing rules, Qarib Publications Establishment, Cairo (in Arabic).

Pols, L., Jekosch, U., 1997. A structured way of looking at the performance of text-to-speech systems. In: Van Santen, J., Sproat, R., Olive, J., Hirschberg, J. (Eds.), Progress in Speech Synthesis. Springer, New York.

Sampson, G., 1985. Writing Systems. Hutchinson, London.

Schmidt, M., Fitt, S., Scott, C., Jack, M., 1993. Phonetic transcription standards for European names (ONOMASTICA). In: Proceedings of the Eurospeech, vol. 93. ESCA, Berlin, pp. 279–282.

Sejnowski, T., Rosenberg, C.R., 1987. Parallel networks that learn to pronounce English text. Complex Systems 1, 145–168.

Silverman, K., Basson, S., Levas, S., 1990. Evaluating synthesizer performance: is segmental intelligibility enough? In: Proceedings of the International Conference on Speech and Language Processing (ICSLP) 90, vol. 2, pp. 981–984.

Sproat, R. (Ed.), 1998. Multilingual Text-to-Speech Synthesis: The Bell Labs Approach. Kluwer Academic Publishers, Dordrecht.

Thorndike, E., Lorge, I., 1944. The Teachers' Word Book of 30,00 Words, Teachers' College, Columbia University, New York.

Van Leeuwen, 1993. Speech maker formalism: a rule formalism operating on a multilevel synchronized data structure. Computer Speech and Language 4, 149–167.

Van Santen, J., 1993. Perceptual experiments for diagnostic testing of text-to-speech systems. Computer Speech and Language 7, 49–100.

Vitale, T., 1991. An algorithm for high accuracy name pronunciation by parametric speech synthesizer. Computational Linguistics 17, 257–276.

University of Umm Alqura, 1970. The Macca list of frequently used Arabic words. Compiled by the Institute of Arabic Language Studies (research unit), University of Umm Alqura, Macca, Saudi Arabia, Safa Publications, Macca.

Yvon, F., 1996. Grapheme-to-phoneme conversion of multiple unbounded overlapping junks. CMP-LG. Paper No. 9608006.

Yvon, F., 1997. Paradigmatic cascades: a linguistically sound model of pronunciation by analogy. In: Proceedings of 35th Annual Meeting of the Association for Computational Linguistics, Madrid, Spain, pp. 429–435.

Yvon, F. et al., 1998. Objective evaluation of grapheme to phoneme conversion of text-to-speech synthesis in French. Computer Speech and Language 12, 393–410.