

KACST Arabic Diacritizer

Mansour Alghamdi¹, Zeeshan Muzafar²

1. King Abdulaziz City for Science and Technology (KACST), Riyadh. 2. King Fahad University of Petroleum and Minerals, Dammam, Saudi Arabia

Keywords: Arabic text diacritization vocalization automatic letters diacritics .

Abstract This paper talks about an innovative Arabic diacritizer (KAD). The diacritizer is solely developed at KACST. Most of the previous published work on diacritization depends on other tools such as Hidden Markov Model Toolkit and morphological analyzer. The present system is small in size, fast in processing and independent from other tools and linguistic rules. KAD uses a stoical method that relies on a quad-gram probability. Its accuracy rate without using any additional methods such as a morphological and syntactical analyzer is relatively high compared to the previous system cited in the literature below.

1. Introduction

As in any other language, Arabic sound system consists of vowels and consonants. Arabic possesses 3 short vowels and 3 long counterparts. The number of consonants is 28 single (Table 1) and 28 geminate counterparts. All of these sounds have representations in the Arabic orthography. The Arabic orthography consists of 35 letters (Table 1 and 2) and 13 diacritics (Table 3). The letters are mainly to represent the consonants. The diacritics are mainly to represent the vowels. When an Arabic text is fully diacritized, every letter must be followed by a diacritic except some letters in certain context (Table 4).

Table 1. Arabic orthography (AO) and their representations in International Phonetic Alphabet (IPA) as the consonant inventory of Modern Standard Arabic.

AO	IPA	AO	IPA	AO	IPA	AO	IPA
	b		ð		t ^ʕ		l
	t		r		ð ^ʕ		m
	θ		z		ʕ		n
	ʒ		s		ʁ		h
	h		ʃ		f		w
	χ		s ^ʕ		q		j
	d		d ^ʕ		k		ʔ

Table 2. Additional Arabic orthographic symbols (AO) and their IPA representations.

AO	IPA	AO	IPA
	a	ʔ	
	ʔ	ʔ	
	ʔ		
	/ʔa/ "utterance initial as in " ""		
	/a/ "preceded by /a/ within a word as in " ""		
	Ø "word initial but not utterance initial as in " ""		
	/h / "utterance final as in " "" and /t/ "else as in " "		

Table 3. Arabic diacritics. The horizontal line represents an Arabic letter.

Diacritic	Definition
	Fathah; represents the low vowel /a/.
	Dhammah; represents the high back vowel /u/.
	Kasrah; represents the high front vowel /i/.
	Tanween Fath; /-an/ comes as word final.
	Tanween Dham; /-un/ comes as word final.
	Tanween Kasr; /-in/ comes as word final.
	Sukoon; the preceding consonant is neither followed by a vowel nor geminate.
	Shaddah and fathah; the preceding consonant is geminate followed by fathah.
	Shaddah and dhammah; the preceding consonant is geminate followed by dhammah.
	Shaddah and kasrah; the preceding consonant is geminate followed by kasrah.
	Shaddah and tanween fathah; the preceding consonant is geminate followed by tanween fathah.
	Shaddah and tanween dhammah; the preceding consonant is geminate followed by tanween dhammah.
	Shaddah and tanween kasrah; the preceding consonant is geminate followed by tanween kasrah.

Table 4. Arabic letters that are not diacritized..

Diacritic	Definition
	<i>Alif mamdoudah</i> is diacritized only when it is word initial and not part of the definite article <i>all</i> .
	<i>Alif maqsourah</i> is always undiacritized.
	<i>Waw</i> is undiacritized when it is part of the long vowel /u:/.
	<i>Yaa</i> is undiacritized when it is part of the long vowel /i:/.
	<i>Lam</i> is undiacritized when it is <i>lam shamsyyah</i> .

It is very rare to use diacritics in modern Arabic text. Newspapers, books and the Internet have Arabic content that is usually written without diacritics. To make the issue close to the reader of English, a sentence such as this, “The man did not mean this book is mine”. would be written as, “Th mn dd nt mn ths bk s mn”. Although this minimizes the representations of the spoken utterances in the text, it requires more efforts from the reader. The reader has to analyze the text morphologically, syntactically and semantically before reading it i.e., restoring the diacritics. To give an example of an Arabic word and how frequent it is diacritized in modern writing, we

searched for the word “ورق” with different diacritics using the Google search engine and got the results in Table 5. Similar results were obtained when searching for other Arabic words. It seems that the diacritized Arabic words on the Internet are less than 1% of the total Arabic content.

Table 5. Statistics of the occurrence of the Arabic word “ورق” with different diacritics on the Internet using the Google search engine.

Arabic Word	English Meaning	Frequency	Percentage
ورق	“without diacritics”	1,380,000	99.91
وَرَقَ	paper	962	00.07
وَرَق	silver	258	00.02
وَرَّقَ	coming out leaves	1	00.00
Total		1,381,221	100

Although undiacritized Arabic text is sufficient for Arabic speakers to use it in writing and reading, this is not the case when dealing with software systems. For example, an Arabic text-to-speech system would not produce speech from undiacritized Arabic text because there is more than one way of saying the same undiacritized written Arabic word. Moreover, when searching for an Arabic word, many unrelated words would be included in the results. This suggests the need to diacritize Arabic text. Another reason for the diacritization is to permit the use of dictionaries and machine translation from and to Arabic. For these reasons and many others, software companies that deal with Arabic realize the importance to develop a system for diacritizing the Arabic text. There are a few systems that are available in the market [1, 2, 3]. However, they are not open source and usually are integrated with other systems. Researchers who are interested in this area have tried their own methods [4, 5, 6, 7, 8, 9]. Hidden Markov Model (HMM) is used in some experiments [4, 7, 9]; morphological rules and acoustic patterns are utilized in another study [5]; a statistical method is used at the word and character levels to predict the diacritics [6]. The results of the previous studies give accuracy rate 74% to 96%. However, the accuracy rate above 80% occurs when a combination of more than two methods are applied, say morphological rules and HMM.

This paper presents an innovative diacritizer that is independent from other tools such as Hidden Markov Model Toolkit (HTK). The innovative diacritizer was developed at King Abdulaziz City for Science and Technology (KACST) under the name KACST Arabic Diacritizer (KAD).

2. Methods and Procedures

The technique used in KAD has two major steps. The first step is to create a very rich list of frequently used Arabic quad-grams (pattern of 4 consecutive diacritized letters). The second step is to use this list in diacritizing almost any Arabic text.

We used the KACST diacritized Arabic text corpus (KDATD) developed by KACST speech team [10] to create the quad-gram list. The KDATD consists of 231 text files representing 22 subjects. Each file has an average of 1000 diacritized words. The quad-grams in KDATD were extracted with their frequencies. The space between the words

is also considered in the quad-grams. The quad-grams having the same letter sequence but different diacritics were grouped. The probability of occurrence for each member of a particular group was computed. The quad-gram with the highest frequency was selected from each group. The probability of a quad-gram with the highest frequency of letters and diacritics combination is calculated as follows:

(س ع د), F_h is the frequency of the highest quad-gram sequence of letters and diacritics and F_t is the total frequency of the quad-gram sequence of letters with different diacritic combinations.

Table 6. A sample of the quad-grams with their probabilities where Li, Di and S stand for letter, diacritic and space, respectively.

A system was developed to diacritize undiacritized Arabic text using the quad-gram database. The input to the system are the sequences of words i.e., sentences. The system assumes each sentence as a sequence of undiacritized letters. The objective is to diacritize the given sequences of undiacritized letters by analyzing the previously computed diacritized quad-grams database.

Now, for each quad-gram shown above, we try to search the corresponding diacritized quad-gram in the database. At this stage, for the sake of simplicity, we assume that all the sequences are found. We extract all these sequences and list them as follows:

$L_2, D_8, L_3, D_9, L_4, D_{10}, L_5, D_{11}, P_3$
 $L_3, D_{12}, L_4, D_{13}, L_5, D_{14}, L_6, D_{15}, P_4$
 $L_4, D_{16}, L_5, D_{17}, L_6, D_{18}, L_7, D_{19}, P_5$
 $L_5, D_{20}, L_6, D_{21}, L_7, D_{22}, L_8, D_{23}, P_6$
 $L_6, D_{24}, L_7, D_{25}, L_8, D_{26}, S, P_7$

Where, P is the probability of the occurrence of a particular quad-gram and D is the diacritic associated with each letter.

The next step is to diacritize each Arabic letter one by one by considering all the consecutive quad-gram sequences, which contain that particular letter. It must be noted here that there can be at most 4 diacritized quad-grams for a particular letter. Now for each letter, all the occurrences in the consecutive quad-grams are considered such that the probabilities of the occurrences having the same diacritic are summed together. Finally, the diacritic associated with quad-gram(s) having the highest aggregated probability is selected for that letter. The flow of the selection of quad-grams for each letter can be understood more clearly from Figure 1. In the figure, a ray is extended diagonally downwards from the first occurrence of each letter in the diacritized quad-grams list that cuts all the occurrences of this letter in the following quad-grams. This way the ray marks all the quad-grams that are processed (as discussed above) for each letter.

In case, if any diacritized quad-gram sequence is not available in the database then a dummy sequence is inserted in its place (this is done just to provide ease in processing). The aforementioned procedure is applied on the available sequences for each letter. This procedure is rewritten mathematically as follows:

p_k – probability of the k^{th} sequence.

p_k^d – probability of the k^{th} sequence with diacritic d.

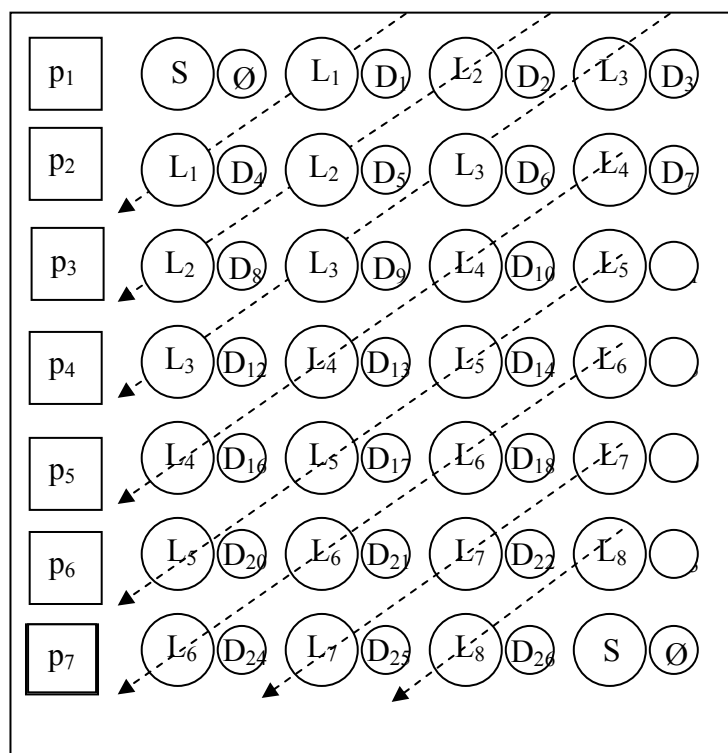
$P(G^d)$ – Aggregate probability of all the sequences with diacritic d. G^d stands for group for diacritic d.

$P(G^d) = \sum_{k=1}^t p_k^d$ t – number of sequences having diacritic d.

$I = \arg \max_{i=1 \dots n} (P(G^d(i)))$ n – number of groups with distinct diacritics.

Selected diacritic (sd) = diacritic ($G^d(I)$)

Figure1. This figure shows the flow of selection of quad-grams and the diacritic for each letter (L).



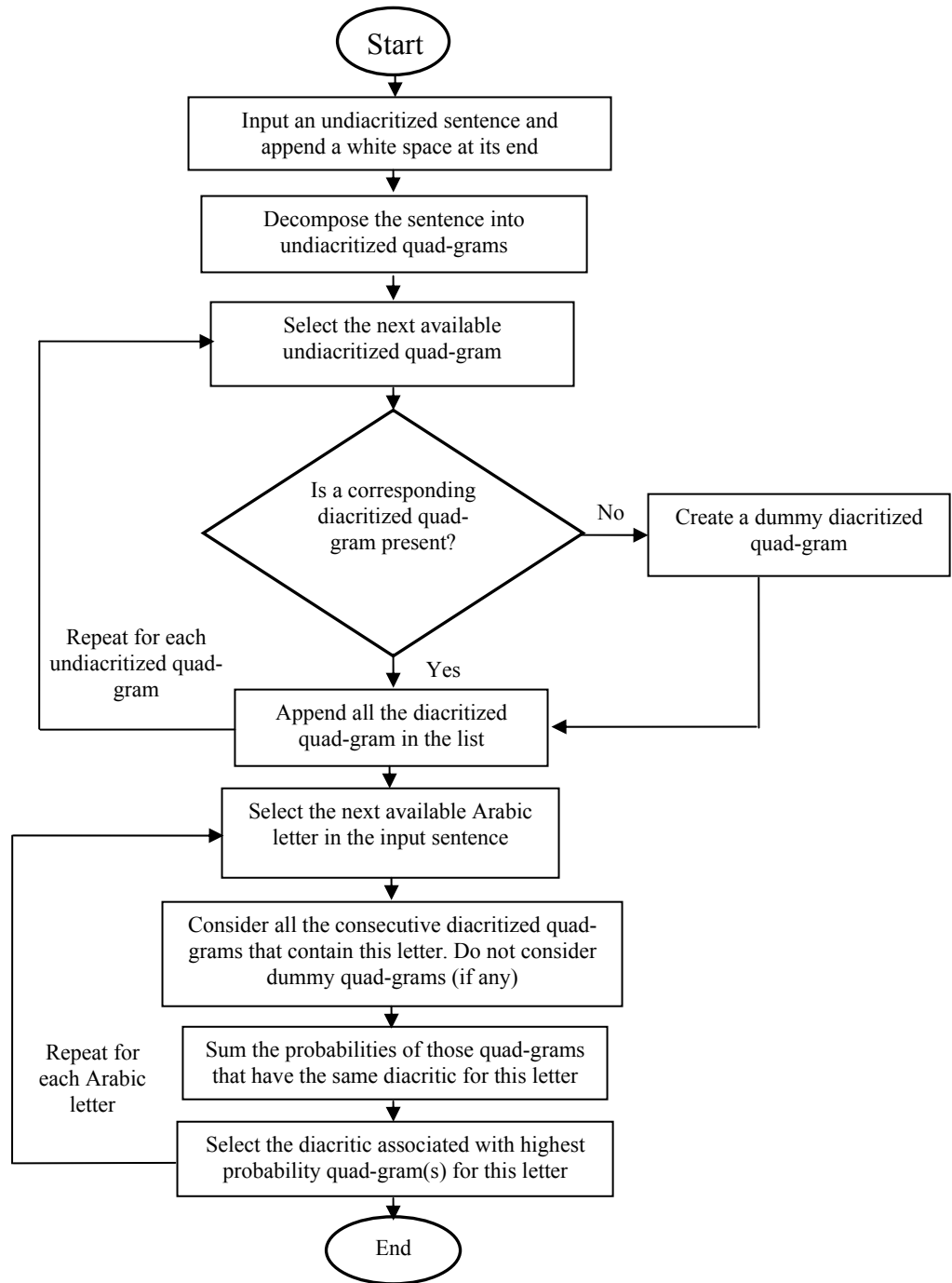
3. Results

The result is a system that is schematized in Figure 2. The size of KAD is less than 3 MB's. Its speed is more than 500 words per second. The system is available for researchers on this link: <http://www.mghamdi.com/KAD.zip>.

Table 7. The results of KAD tests.

File Name	Word No	Letter No	Error Rate
A6	1000	4339	9.04%
A30	1005	4900	8.02%
A62	1019	5351	7.05%
R21	974	5261	6.50%
R50	1019	5012	7.04%
Total	5017	24863	7.64%
T01	727	3364	6.93%
T02	374	1917	9.65%
T03	328	1506	9.50%
T04	253	1390	7.77%
T05	547	2611	12.33%
T06	390	1961	8.98%
T07	277	1496	5.62%
T08	209	1051	6.71%
T09	334	1672	5.46%
T10	361	1684	9.87%
Total	3800	18652	8.87%
Grand Total	8817	43515	8.52%

Figure 2. A schematic drawing of KAD.



To test the KAD's accuracy rate, two sets of data were selected. The first set represents 5 articles taken from KDATD. They were undiacritized and later diacritized by KAD. The second set is 10 articles taken from Alriyadh Newspaper. The articles are under different topics and written by different authors. They were diacritized manually and by KAD. The test of KAD diacritization is for all the letters including word final. The results of KAD diacritization were compared to those of human and the results are shown in Table 7. The average error rate of the text that was taken from KDATD is 7.64% and for Alriyadh Newspaper is 8.87. The average error rate of both text sets is 8.52. This error rate is lower than any error rate reported in the literature.

4. Conclusions

This paper presents KAD as an innovative system for diacritization. KAD applies a new methodology that is not tied to other toolkits such as HTK. The initial evaluation of the system is encouraging. However, the accuracy rate can be improved further by adding the other possible quad-grams that are not included in KDATD. Moreover, linguistic information can be fed into the system to add morphological and syntactic rules that can enhance the accuracy rate.

Acknowledgements

This paper is supported by KACST.

References

- [1] Sakhr: <http://www.sakhr.com/>
- [2] Research & Development International (RDI): <http://www.rdi-eg.com>
- [3] Cimos: <http://www.cimos.com>
- [4] Gal, Ya'akov (2002) An HMM Approach to Vowel Restoration in Arabic and Hebrew. In Proceedings of the Workshop on Computational Approaches to Semitic Languages. Philadelphia. 27-33
- [5] Dimitra, Vergyri and Katrin, Kirchho (2004) Automatic diacritization of Arabic for Acoustic Modeling in Speech Recognition. In Proceedings of the 20th International Conference on Computational Linguistics, Geneva, 66-73.
- [6] Ananthakrishnan, Sankaranarayanan, Shrikanth S. Narayanan and Srinivas Bangalore (2005) Automatic Diacritization of Arabic Transcripts for Automatic Speech Recognition. In Proceedings of International Conference On Natural Language Processing, Kanpur, India.
- [7] Elshafei, Mustafa, Husni Almuhtasib and Mansour Alghamdi (2006). Machine Generation of Arabic Diacritical Marks. The 2006 World Congress in Computer Science Computer Engineering, and Applied Computing. Las Vegas, USA. 26-29/6/2006.
- [8] Elshafei, Mustafa, Husni Almuhtasib and Mansour Alghamdi (2006) Statistical Methods for Automatic Diacritization of Arabic Text. The Saudi 18th National Computer Conference. Riyadh. 18: 301-306.
- [9] Nelken, Rani and Stuart M. Shieber (2005) Arabic Diacritization Using Weighted Finite-State Transducers. In Proceedings of the Workshop on Computational Approaches to Semitic Languages Workshop, University of Michigan, Ann Arbor. 79-86.
- [10] Alghamdi, Mansour, Muhammad Khursheed, Mustafa Elshafei, Fayz Alhargan, Muhammed Alkanhal, Abu Aus Alshamsan, Saad Alqahtani, Syed Zeeshan Muzaffar, Yasser Altowim, Adnan Yusuf, Husni Almuhtasib (2006) Automatic Arabic Text Diacritizer. Final Report, KACST: CI.25.02.