Automatic restoration of Arabic diacritics: A simple, purely statistical approach

Article *in* Arabian Journal for Science and Engineering · December 2010

CITATIONS
27

READS
561

2 authors, including:

Mansour Alghamdi
National Center for Assessment
58 PUBLICATIONS **794** CITATIONS

SEE PROFILE

AUTOMATIC RESTORATION OF ARABIC DIACRITICS: A SIMPLE, PURELY STATISTICAL APPROACH

Mansour Alghamdi*

King Abdulaziz City for Science and Technology (KACST), Riyadh, Saudi Arabia

Zeeshan Muzaffar

Department of Computer Science, University of Western Ontario, Canada

and Hazim Alhakami

King Abdulaziz City for Science and Technology (KACST), Riyadh, Saudi Arabia

الخلاصة:

تعرض هذه الورقة طريقة مبتكرة لتشكيل النص العربي آليا. حيث تعتمد معظم الطرق السابقة المنشورة على أدوات أخرى كأدوات أنموذج ماركوف الخفي و/أو المعرفة اللغوية كالصرف والنحو. والنظام الذي بين أيدينا هو نظام أساسي يتميز بصغر حجمه وسرعته في المعالجة بمعزل عن الأدوات الحاسوبية المعروفة أو القوانين اللغوية. إذ يستخدم هذا النظام طريقة إحصائية بحتة تعتمد على احتمالية التسلسل الرباعي للحروف. وينتج عنه درجة عالية في دقة التشكيل إذا ما قورن بالأنظمة السابقة التي تعتمد على الإحصاء فقط بوصفه نظاماً أساسياً .

*Corresponding Author:
E-mail: mghamdi@kacst.edu.sa

Paper Received May 25, 2010; Paper Revised October 9, 2010; Paper Accepted October 30, 2010

ABSTRACT

This paper is about an innovative Arabic diacritizer. Most of the previous published works on diacritization depends on other tools such as Hidden Markov Model Toolkit and/or higher linguistic knowledge such as morphology and syntax. The present system is a baseline system which is small in size, fast in processing, and independent from other tools and linguistic rules. The system uses a statistical method that relies on quad-gram probabilities. Its accuracy rate is relatively high when compared to previous systems that are based only on statistics.

***Key words*:** Automatic Arabic diacritization, Arabic vocalization, letter diacritics

AUTOMATIC RESTORATION OF ARABIC DIACRITICS: A SIMPLE, PURELY STATISTICAL APPROACH

1. INTRODUCTION

As in any other language, the Arabic sound system consists of vowels and consonants. Arabic possesses 3 short vowels and 3 long counterparts. The short vowels are /i, u, a/ and they are phonetically half the duration of their long counterparts /i:, u:, a:/. As for consonants, there are 28 singles (Table 1) and their geminate counterparts. The difference between single and geminate Arabic consonants is mainly the duration; a single consonant is almost half the duration of its counterpart. Each Arabic sound has a symbol in the Arabic orthography. The Arabic orthography consists of 35 letters (Tables 1 and 2) and 13 diacritics (Table 3). The letters are mainly to represent the consonants. The diacritics are mainly to represent the vowels and gemination. When an Arabic text is fully diacritized, every letter must be followed by a diacritic except some letters in certain contexts (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
ب	b	ذ	ð	ط	$t^ˤ$	ل	l
ت	t	ر	r	ظ	$ð^ˤ$	م	m
ث	θ	ز	z	ع	ʕ	ن	n
ج	dʒ	س	s	غ	ɣ	هـ	h
ح	ħ	ش	ʃ	ف	f	و	w
خ	χ	ص	$s^ˤ$	ق	q	ي	j
د	d	ض	$d^ˤ$	ك	k	ء	ʔ

Table 2. Additional Arabic Orthographic Symbols (AO) and Their IPA Representations

AO	IPA	AO	IPA
ى	a	إ	ʔ
آ	ʔaː	ئ	ʔ
أ	ʔ	ؤ	ʔ
ا	/ʔa/ utterance initial as in "العلم"		
	/a/ preceded by /a/ within a word as in "عالم"		
	∅ word initial but not utterance initial as in "في العلم"		
ة	/h / utterance final as in "سماء صافية" and /t/ elsewhere as in "معرفة الإنسان"		

Table 3. Arabic Diacritics. The horizontal line represents an Arabic letter. *Shaddah* always comes with another diacritic, so, they are treated in our system as one unit.

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

Table 4. Arabic Letters that are not Diacritized

Diacritic	Definition
ا	*Alif mamdoudah* is diacritized only when it is word initial and not part of the definite article *all*.
آ ، ى	*Alif maqsourah (ى) and Alif madd (آ)* are always undiacritized.
و	*Waw* is undiacritized when it is part of the long vowel /u:/.
ي	*Yaa* is undiacritized when it is part of the long vowel /i:/.
ل	*Lam* is undiacritized when it is part of the definite article *lam shamsyyah*.

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 more understandable to the reader of English, a sentence such as "The man did not mean this book is mine" would be written as, "Th mn dd nt mn ths bk s mn". The sequence "mn" represents three different words. Similar sequences exist in modern written Arabic. It is true that similar sets of words exist in different languages, including English, such as "lead" and "read", which have different pronunciations according to the verb tense, but the number of these sets is not as high as that of the undiacritized words in Arabic.

Although undiacritized Arabic orthography minimizes the representations of the spoken utterances in the text, it requires more effort from the reader. The reader has to analyze the text morphologically, syntactically, and semantically before reading it in order to restore 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. They show that the diacritized Arabic words on the Internet are less than 1% of the total Arabic content.

Table 5. Statistics on 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
ورق	paper	962	00.07
ورِق	silver	258	00.02
وَرَق	leaves coming out	1	00.00
Total		1,381,221	100

Although undiacritized Arabic text is sufficient for Arabic speakers to use 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 of developing a system for diacritizing the Arabic text. There are a few systems that are available on the market [1,2,3]. However, they are not open source and usually integrated with other systems.

Researchers who are interested in this area have tried their own methods [4–9]. The Hidden Markov Model (HMM) is used in some experiments [4,7,9]; morphological rules and acoustic patterns are utilized in another study [5]; and, a statistical method is used at the word and character levels to predict the diacritics [6]. The results of the previous studies show an accuracy rate of 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.

Several related approaches have lately been reported. Zitouni *et al.* [11] used the Arabic Treebank database, distributed by Linguistic Data Consortium (LDC) [12], in a system based on an array of lexical, segment-based, and part-of-speech tag features to train and test the system. They reported a 5.5% error rate at letter level, including word final, when combining all the features in the array. Diab *et al.* [13] also used Arabic Treebank to test different schemes ranging from full to partial diacritization by applying a trigram language model. They found that the presence of more diacritics in the text does not improve the outcome of their statistical machine translation. Ananthakrishnan *et al.* [5] used the Treebank to train and test their diacritization system. They used a 4-gram character model in addition to the morphological analyzer. They got an accuracy rate of 74.8% at the word level that was present in the training set but only 48.2% at the new words. Nelken and Shieber [9] applied a trigram language model at the word level using the Treebank. 90% of the database was used for training and 10% for testing. Several procedures were used, including clitic concatenation, trigram, and quad-gram to arrive at a 12.79% error rate. Habash and Rambow [14] combined a tagger and a lexeme language model to diacritize Arabic letters. They got an error rate of 4.8% at character level for all letters using Treebank.

This paper presents an innovative diacritizer that is independent from other tools such as the Hidden Markov Model Toolkit (HTK). It is used as a baseline without the support of other linguistic factors such as morphology, syntax, and lexicon.

2. METHODS AND PROCEDURES

The technique used in the present system 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 a 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 that have 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 probability was selected from each group. The probability of a quad-gram with the highest frequency of letters and diacritics combination is calculated as follows:

$$P = \frac{F_h}{F_t}$$

where P is the probability of a certain quad-gram sequence of letters (such as: "س ا 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.

The result is a list (database) of 68378 quad-grams that have the highest quad-gram probabilities. Table 6 lists some quad-grams from the database with their probabilities. The list is only 4% of the quad-grams that Arabic letters theoretically generate (35 letters + 1 space).

Table 6. A Sample of the Quad-Grams With Their Probabilities Where Li, Di and S Stand for Letter, Diacritic, and Space, Respectively

L_1+D_1	L_2+D_2	L_3+D_3	L_4+D_4	Pro.
دْ	ك	ي	ر	0.57
دُ	لْ	S	أَ	0.33
دْ	لْ	S	إِ	1.00
دْ	لْ	S	ا	0.40

The system was developed to diacritize undiacritized Arabic text using the quad-gram database. The input to the system is 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 applying the previously computed diacritized quad-grams database.

Consider an example, where we have the undiacritized letter sequence L1, L2, L3, L4, L5, L6, L7, L8 as an input. We append a single white space (S) at both ends of this sequence. The sequence is then decomposed into quad-grams as follows:

S, L_1, L_2, L_3
L_1, L_2, L_3, L_4
L_2, L_3, L_4, L_5
L_3, L_4, L_5, L_6
L_4, L_5, L_6, L_7
L_5, L_6, L_7, L_8
L_6, L_7, L_8, S

Now, for each quad-gram shown above, we 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:

S, $\mathbf{L_1}$, D_1, $\mathbf{L_2}$, D_2, $\mathbf{L_3}$, D_3, P_1

$\mathbf{L_1}$, D_4, $\mathbf{L_2}$, D_5, $\mathbf{L_3}$, D_6, $\mathbf{L_4}$, D_7, P_2

$\mathbf{L_2}$, D_8, $\mathbf{L_3}$, D_9, $\mathbf{L_4}$, D_{10}, $\mathbf{L_5}$, D_{11}, P_3

$\mathbf{L_3}$, D_{12}, $\mathbf{L_4}$, D_{13}, $\mathbf{L_5}$, D_{14}, $\mathbf{L_6}$, D_{15}, P_4

$\mathbf{L_4}$, D_{16}, $\mathbf{L_5}$, D_{17}, $\mathbf{L_6}$, D_{18}, $\mathbf{L_7}$, D_{19}, P_5

$\mathbf{L_5}$, D_{20}, $\mathbf{L_6}$, D_{21}, $\mathbf{L_7}$, D_{22}, $\mathbf{L_8}$, D_{23}, P_6

$\mathbf{L_6}$, D_{24}, $\mathbf{L_7}$, D_{25}, $\mathbf{L_8}$, 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 (L).

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 by summing the probabilities of the occurrences of the same diacritic for each letter. 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-gram 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.

If a 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). For example, if the intended sequence is: ق ب ج د ن ل م ف ص ط , then the expected quad grams would be: ق ب ج د - ب ج د ن - ج د ن ل - د ن ل م - ن ل م ف - ل م ف ص – م ف ص ط . However, suppose that the second quad gram - ب ج د ن - is not available in the database. In such a case, the system would insert a blank quad gram instead and continue with the remaining available quad grams.

The aforementioned procedure is applied to the available sequences for each letter (L). This procedure is rewritten mathematically as follows:

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

$p_k^{d_q}$ – probability of the k^{th} sequence with q^{th} candidate diacritic, *i.e.*, d_q. There can be at most 4 different candidate diacritics for any L.

$P(G^{d_q})$ – Aggregate probability of all the sequences with d_q. G^{d_q} stands for group for d_q, *i.e.*, all sequences that have d_q.

$P(G^{d_q}) = \sum_{r=1}^{t} p_r^{d_q}$ t – number of sequences having d_q.

$I = \arg\max_{q=1\ldots n}(P(G^{d_q}))$ n – number of candidate groups.

Selected diacritic (d_s) = d_I

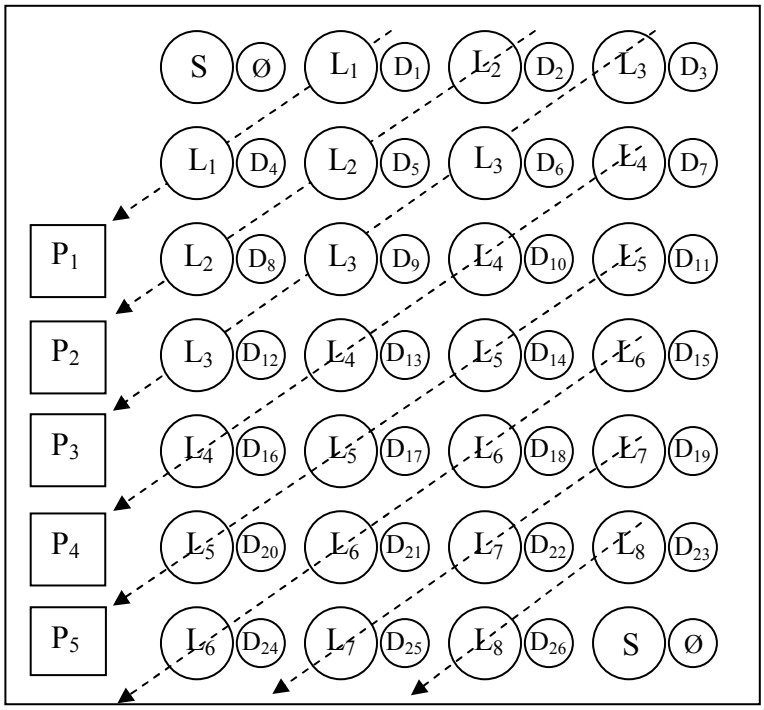

Figure 1: The flow of selection of quad-grams and the diacritic (D) for each letter (L)

3. RESULTS

The result is a system that is schematized in Figure 2. The size of the system is less than 3 MB's. Its speed is more than 500 words per second using a 533-MHz processor. The system is available for researchers on this link: http://www.mghamdi.com/KAD.zip.

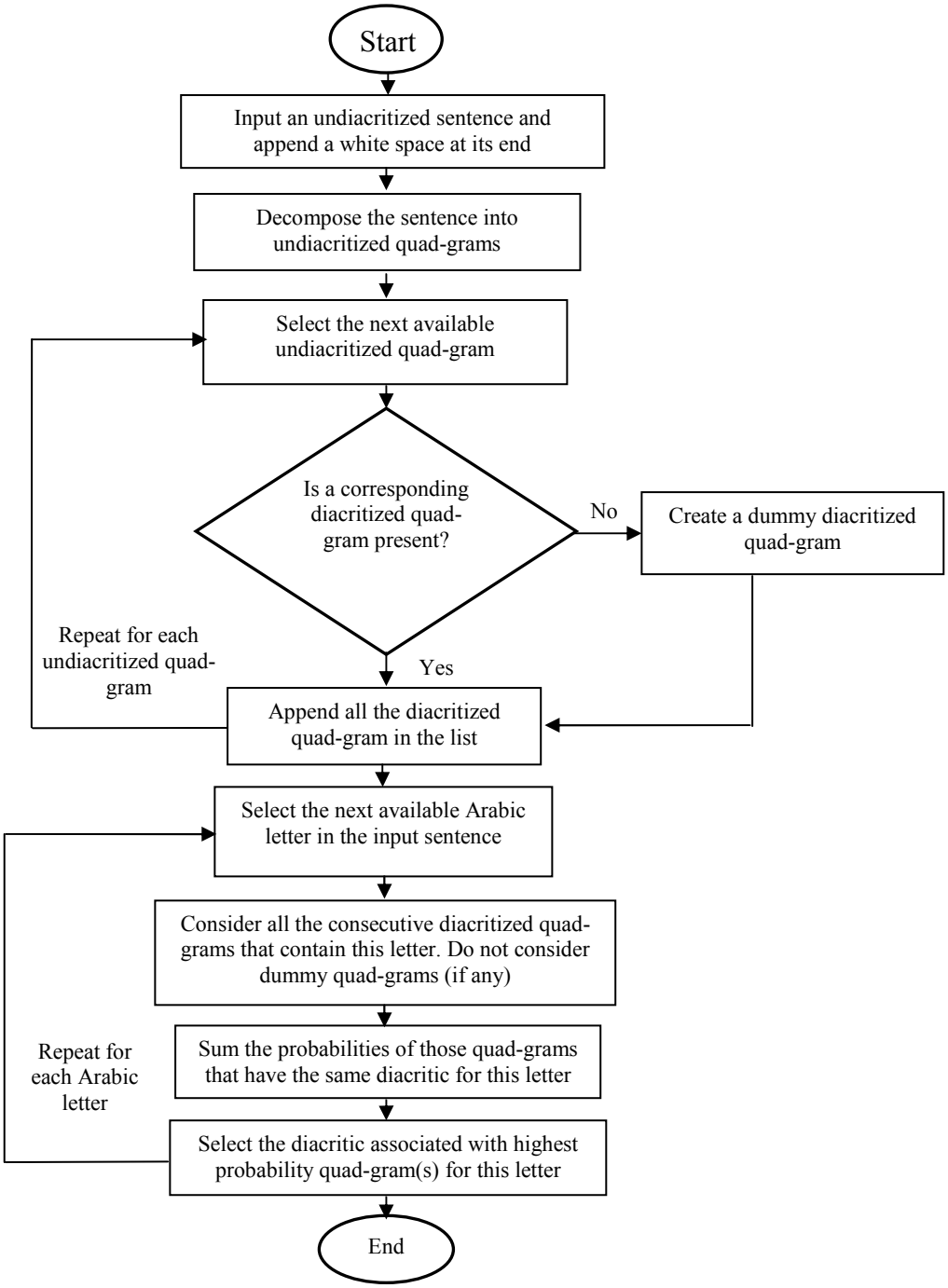

Figure 2: A schematic drawing of the system

Previous results of testing the system [15] show a 7.64% error rate when the test set (5017 words) is from the training set and 8.52% when the test set (3800 words) is not included in the training set. The results are based at character level of all letters including word final.

To test the system on a known database, the first 64 files of the LDC test set were selected (Appendix 1). They include 15983 words with 83897 letters. However, we found several deficiencies in this set: 1) a large portion of the letters (more than 10%) are undiacritized: the final *hamzah*, the letter before *alif*, foreign words, *allamu alqamariyah*, the letter after *allamu alshamsiyah,* final letters with *sukoun*, *etc.*, 2) the *tanween* comes after *alif* where it should come before it, 3) *fathah* comes after *alif* in the word *alhayah* where it should be before *alif*, 4) the presence of characters that are not used in modern written Arabic, namely the small *alif* and *hamzatu alwasl*. After these deficiencies were corrected, the set was undiacritized and diacritized by our system. The results are shown in Table 7.

The error rate is relatively higher than that reported in the previous study [15]. This is due to the fact that the testing set is not from the same source as that of the training set. Previous studies had the training and testing sets from the same source, usually LDC. Another reason seems to be the presence of many foreign words, names of politicians and organizations, in the testing set, which is mostly on politics. Table 7 also shows a striking error rate decrease at word level by 44.42% of its original error rate when the word final diacritics are excluded. This result is expected since word final diacritization is linked to Arabic syntax which needs a more intelligent system. However, the error rate reduction at the character level is only 33.12% of the original error rate, which is less than that of word-level error rate.

Table 8 shows the types of errors that occur when using the system. Most of the errors occur when the system replaces a correct diacritic with a wrong one. Fewer errors occur when the system diacritizes a letter which is not supposed to be diacritized (addition) or vice versa (omission).

Table 7. Error Rate at Word and Character Levels With and Without the Diacritization of the Word-Final Letters

Error Rate at Word Level			Error Rate at Character Level		
Including Word Final	**Excluding Word Final**	**Error Reduction**	**Including Word Final**	**Excluding Word Final**	**Error Reduction**
46.83	26.03	44.42	13.83	9.25	33.12

Table 8. Error Rates According to Their Types of Errors With and Without the Diacritization of the Word-Final Letters

Including Word Final			Excluding Word Final		
Replaced	**Addition**	**Omission**	**Replaced**	**Addition**	**Omission**
10.74	1.21	1.88	6.37	1.24	1.64

What makes this method different from the previous ones is it does not stop at one quad-gram in diacritizing a certain letter. It combines up to 4 different quad-grams, *i.e.*, 7 sequential letters, which raises the accuracy probability since the more sequential letters, the more the accuracy of the diacritization is. As it is known, if the diacritics of the letters of the whole sentence are known, we will have error-free diacritization. By the same token, if we rely on mono-gram probability, the accuracy rate would be very low. In addition, the system does not work on a predefined word list; it can diacritize any Arabic words regardless of whether they are in the training set or not. Moreover, the system would work even when certain quad grams are not in the training list. For example, if the second quad-gram in Figure 1 was not on the list, the system would be able to diacritize the letters based on the other available sequences. Other advantages of the system are its speed, size, and availability online for researchers.

4. CONCLUSIONS

This paper presents a system for diacritizing Arabic text which is innovative and shows high performance. It 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 syntactical rules that can enhance the accuracy rate.

ACKNOWLEDGMENTS

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] Y. Gal, "An HMM Approach to Vowel Restoration in Arabic and Hebrew", in *Proceedings of the Workshop on Computational Approaches to Semitic Languages*2002. Philadelphia, pp. 27–33.

[5] V. Dimitra and K. Katrin, "Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition", in *Proceedings of the 20th International Conference on Computational Linguistics* 2004. Geneva, pp. 66–73.

[6] S. Ananthakrishnan, S. S. Narayanan, and S. Bangalore, "Automatic Diacritization of Arabic Transcripts for Automatic Speech Recognition", in *Proceedings of International Conference on Natural Language Processing* 2005. Kanpur, India.

[7] M. Elshafei, H. Almuhtasib, and M. Alghamdi, "Machine Generation of Arabic Diacritical Marks", *The 2006 World Congress in Computer Science Computer Engineering, and Applied Computing* 2006. Las Vegas, USA.

[8] M. Elshafei, H. Almuhtasib, and M. Alghamdi, "Statistical Methods for Automatic Diacritization of Arabic Text", *The Saudi 18th National Computer Conference* 2006. Riyadh. **18**, pp. 301–306.

[9] R. Nelken and S. T. M. Shieber, "Arabic Diacritization Using Weighted Finite-State Transducers", in *Proceedings of the Workshop on Computational Approaches to Semitic Languages Workshop* 2005. University of Michigan, Ann Arbor, pp. 79–86.

[10] M. Alghamdi, M. Khursheed, M. Elshafei, F. Alhargan, M. Alkanhal, A. A. Alshamsan, S. Alqahtani, S. Z. Muzaffar, Y. Altowim, A. Yusuf, and H. Almuhtasib, "Automatic Arabic Text Diacritizer", Final Report, KACST: CI.25.02.

[11] I. Zitouni, J. S. Sorensen, and R. Sarikaya, "Maximum Entropy Based Restoration of Arabic Diacritics", in *Proceedings of ACL'06* 2006.

[12] Linguistic Data Consortium: http://www.ldc.upenn.edu

[13] M. Diab, M. Ghoneim, and N. Habash, "Arabic Diacritization in the Context of Statistical Machine Translation", *MT Summit XI*, 10-14 September 2007, Copenhagen, Denmark. Proceedings, pp.143–149.

[14] N. Habash and O. Rambow, "Arabic Diacritization Through Full Morphological Tagging", in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)* 2007. Rochester, New York.

[15] M. Alghamdi and Z. Muzaffar, "KACST Arabic Diacritizer", *The First International Symposium on Computers and Arabic Language* 25-28/3/2007.

APPENDIX I

UMAAH_UM.ARB_20010721-e.0018.sgm

UMAAH_UM.ARB_20010721-e.0023.sgm

UMAAH_UM.ARB_20020106-a.0019.sgm

UMAAH_UM.ARB_20020106-a.0024.sgm

UMAAH_UM.ARB_20020113-a.0001.sgm

UMAAH_UM.ARB_20020113-a.0002.sgm

UMAAH_UM.ARB_20020113-a.0004.sgm

UMAAH_UM.ARB_20020113-a.0010.sgm

UMAAH_UM.ARB_20020113-a.0020.sgm

UMAAH_UM.ARB_20020113-e.0008.sgm

UMAAH_UM.ARB_20020120-a.0001.sgm

UMAAH_UM.ARB_20020120-a.0002.sgm

UMAAH_UM.ARB_20020120-a.0004.sgm

UMAAH_UM.ARB_20020120-a.0005.sgm

UMAAH_UM.ARB_20020120-a.0006.sgm

UMAAH_UM.ARB_20020120-a.0007.sgm

UMAAH_UM.ARB_20020120-a.0009.sgm

UMAAH_UM.ARB_20020120-a.0018.sgm

UMAAH_UM.ARB_20020120-a.0021.sgm

UMAAH_UM.ARB_20020120-e.0005.sgm

UMAAH_UM.ARB_20020120-e.0028.sgm

UMAAH_UM.ARB_20020127-a.0001.sgm

UMAAH_UM.ARB_20020127-a.0002.sgm

UMAAH_UM.ARB_20020127-a.0007.sgm

UMAAH_UM.ARB_20020127-a.0008.sgm
UMAAH_UM.ARB_20020127-a.0009.sgm
UMAAH_UM.ARB_20020127-a.0010.sgm
UMAAH_UM.ARB_20020127-a.0011.sgm
UMAAH_UM.ARB_20020127-e.0007.sgm
UMAAH_UM.ARB_20020127-e.0009.sgm
UMAAH_UM.ARB_20020127-e.0024.sgm
UMAAH_UM.ARB_20020203-a.0005.sgm
UMAAH_UM.ARB_20020203-a.0008.sgm
UMAAH_UM.ARB_20020203-a.0009.sgm
UMAAH_UM.ARB_20020203-a.0011.sgm
UMAAH_UM.ARB_20020203-a.0025.sgm
UMAAH_UM.ARB_20020210-a.0010.sgm
UMAAH_UM.ARB_20020210-a.0013.sgm
UMAAH_UM.ARB_20020210-a.0015.sgm
UMAAH_UM.ARB_20020210-a.0018.sgm
UMAAH_UM.ARB_20020210-a.0019.sgm
UMAAH_UM.ARB_20020210-a.0020.sgm
UMAAH_UM.ARB_20020210-e.0006.sgm
UMAAH_UM.ARB_20020217-a.0002.sgm
UMAAH_UM.ARB_20020217-a.0006.sgm
UMAAH_UM.ARB_20020217-a.0007.sgm
UMAAH_UM.ARB_20020217-a.0009.sgm
UMAAH_UM.ARB_20020217-a.0010.sgm
UMAAH_UM.ARB_20020217-a.0017.sgm
UMAAH_UM.ARB_20020217-a.0019.sgm
UMAAH_UM.ARB_20020224-a.0005.sgm
UMAAH_UM.ARB_20020224-a.0008.sgm
UMAAH_UM.ARB_20020224-a.0010.sgm
UMAAH_UM.ARB_20020224-a.0012.sgm
UMAAH_UM.ARB_20020224-a.0021.sgm
UMAAH_UM.ARB_20020224-e.0014.sgm
UMAAH_UM.ARB_20020224-e.0015.sgm
UMAAH_UM.ARB_20020224-e.0025.sgm
UMAAH_UM.ARB_20020303-a.0020.sgm
UMAAH_UM.ARB_20020303-e.0010.sgm
UMAAH_UM.ARB_backissue_30-e.0015.sgm
UMAAH_UM.ARB_backissue_31-a.0002.sgm
UMAAH_UM.ARB_backissue_31-a.0007.sgm
UMAAH_UM.ARB_backissue_31-a.0010.sgm