A smart Dictionary for the Arabic Full-Form Words

Ali EL-Desoky, Marwa Fayz, Doaa Samir

Abstract— Avery long time is required when searching for a given word in the full-form words dictionary if these words are stored in a single database . A smart enough way is required when designing these type of dictionary in order to decrease the search time as much as possible to speed up the whole process execution time.

This paper introduces a smart dictionary in which the full-form words are distributed into a large number of tables(files). This dictionary distributes the full-form words over 30 tables each of which holds the words start with the same alphabetic character this make the search easier and faster.

Index Terms—Arabic, human language technologies (HLT), hybrid, morphological analysis, morphology, natural language processing (NLP), phonetic transcription, phonological analysis, statistical language model (SLM), statistical, syntax.

I. INTRODUCTION

Arabic is a class of languages where the intended pronunciation of a written word cannot be completely determined by its standard orthographic representation; rather, a set of special diacritics is needed to indicate the intended pronunciation. Different diacritics for the same spelling produce different words with different meanings (e.g. عِلْم"science", عَلَم“flagعَلِمَ “knew” … etc.). In most genres of written Arabic results in widespread ambiguities in pronunciation and (in some cases) meaning. Automatic processing of Arabic is often hampered by the lack of diacritics. Text-to-speech (TTS), Part-Of- Speech (POS) tagging, Word Sense Disambiguation (WSD), and Machine Translation can be enumerated among a longer list of applications that vitally benefit from automatic diacritization [1-4]. The Arabic alphabet consists of 28 letters; 25 letters represent the consonants such as ـب(pronounced as /b/) and 3 letters represent the long vowels such as ـا ,ـى (both ـا,ـى pronounced as /a/),ي (pronounced as /i/), and ـو (pronounced as/u:/), each of them may have eight different shapes and some combinations of them, representing short vowels, nunation (for doubled case endings), and syllabification marks [5-7] . Combinations of the Shadda diacritic with other diacritics may occur such as Shadda-Fatha (pronounced as /b//b//a/), and Shadda-Tanween_Damma (pronounced as /b//b//un /).

One major challenge with Arabic is its rich derivative and inflective nature, so it is very difficult to build a complete vocabulary that covers all (or even most of) the Arabic generable words [8],[9]. In fact, while Arabic has a very rich vocabulary with regard to full-form words, the resulting data sparseness is much more manageable when atomic entities of words (morphemes) are considered separately, due to Arabic is very systematic and rich yet compact morphology [10],[11]. Hence, reliable Arabic morphological analysis is crucial for Arabic diacritization. An Arabic diacritization systems that factorize input Arabic text into all the possible morphemes and case diacritics is proposed. This system statistically disambiguate the most likely sequence of these entities via deep lattice search, hence infer the most likely diacritization and phonetic transcription of the input text [12]. While the system is excellent cover the language; its drawback is that the search space for the correct diacritics using the factored word components is much larger than the original search space of full-form words. This larger search space seems to require larger size of training data, which is expensive and time-consuming to build and validate [11]. Recently been trying the same statistical language modeling and disambiguation methodologies over full-form Arabic words instead of factorized ones is proposed. While the letter approach proved to be faster to run and faster to learn; i.e., it produces more accurate diacritization using the same size of training data, it apparently suffers from the problem of poor coverage. It has then been realized that a hybrid of both approaches may again the advantage of each[10]. All of these approaches uses a single database to hold the whole data this required substantial hardware and software start-up costs and any damage to database affects virtually all application programs and make any search time consuming.

This paper proposes a simple dictionary in which the full-form words are distributed over a large number of small size tables(files) instead of a single one. This way the problems encountered with large-scale database are overcome.

The paper consists of 5 section the first of which is the introduction. Section II gives a brief review of an Arabic morphological analysis & pos tagging models. Section III explains the structure of the proposed dictionaries. Section IV comparison studies. Finally Section V is devoted for conclusion.

II. THE ARABIC FACTORIZATION MODELS

The diacritization of an Arabic word consists of two components; morphology-dependent and syntax-dependent [13],[14]. The morphological diacritization distinguishes different words with the same spelling from one another; e.g. عِلْمwhich means “science” and عَلَم which means “flag”, the syntactic case of the word within a given sentence; i.e. its role in the parsing tree of that sentence, determine the syntax-dependent diacritic of the word. For example; درسْتُ عِلْمَ الرياضيات implies the syntactic diacritic of the target word - which is an “object” in the parsing tree - is “Fatha”, while يفيدُ عِلْمُ الرياضياتِ جميعَ العلوم implies the syntactic diacritic of the target word – which is a “subject” in the parsing tree - is “Damma”.

A. *Arabic Morphological Analysis*

Arabic morphological model assumes the

Manuscript received on November 2012.

Ali EL-Desoky, Department of CSE, University/ Mansoura/EGYPT, India.

Mrwa Fayz, Department of CSE, University/ Mansoura/ EGYPT, India.

Doaa Samir, Department of CSE, University/ Mansoura/ EGYPT, India.

canonical structure uniquely representing any given Arabic word w to be a quadruple of lexemes (or morphemes) so that w→q = (t: p, r, f, s) where p is prefix code, r is root code, f is pattern (or form) code, and s is suffix code. The type code t can signify words belonging to one of the following 4 classes: Regular Derivative (wrd), Irregular Derivative (wid), Fixed (wf), or Arabized (wa) [15-19].Prefixes and suffixes; P and S, the 4 classes applied on patterns giving Frd , Fid , Ff , and Fa, plus only 3 classes applied on roots1; Rd , Rf , and Ra constitute together the 9 categories of lexemes in this model. As shown in figure 1 and table1.

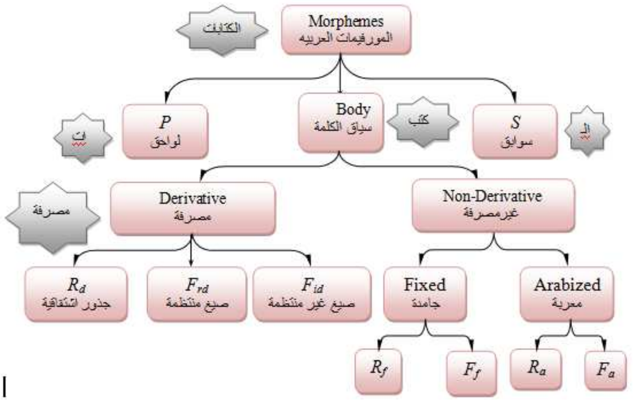


Fig. 1: Classifying the 9 types of morphemes in the Arabiclexicon of ArabMorpho© ver.4.

Table 1. Basic set of arabic diacritics

Suffix	*Pattern*	*Root*	*Prefix*	*Word type*	*Sample word*
-	مَا	اَلَّذِي	فَــ	*Fixed*	فَمَا
ـــه	تَفَاعَلَ	تول	تَــ	*Regular Derivative*	تَتَاوَله
ـــات	كتب	فِعَال	الــ	*Regular Derivative*	اَلْكِتَابَات
ـــيَّة	فِعْل	علم	الــ	*Regular Derivative*	اَلْعِلْمِيَّة
-	مِنْ	مِنْ	-	*Fixed*	مِنْ
-	مَفَاعِيل	وضع	-	*Regular Derivative*	مواضيع
ـــة	مُتَّخَذ	أخذ	-	*Irregular Derivative*	متخذة

B. *Arabic PoS-Tagging*

PoS tagging is a fundamental linguistic analysis process where PoS- tags that convey the basic context-free syntactic features of separate words are extracted Table 2 displayed below shows a sample Arabic POS.

Table 2. Pos-Tags Vectors Of Sample Arabic Words

Sample word	*Arabic PoS tags vector*
فَمَا	[Conjunction, Noun, Relative Pronoun, Null Suffix] [عطف، اسم، اسم موصول، لا لاحقة]
تَتَنَاوَله	[Present, Active, Verb ,Objective Pronoun] [مضارع، مبني للمعلوم، فعل، ضمير نصب]
اَلْكِتَابَات	[Definitive, Noun, Plural, Feminine] [ال التعريف، اسم، جمع، مؤنَّث]
اَلْعِلْمِيَّة	[Definitive, Noun, Relative Adjective, Feminine, Single] [ال التعريف، اسم، نسب، مؤنَّث، مفرد]
مِنْ	[Null Prefix, Preposition, Null Suffix] [لا سابقة، حرف، لا لاحقة]
مَوَاضِيع	[Null Prefix, Noun, No SARF, Plural, Null Suffix] [لا سابقة، اسم، ممنوع من الصرف، جمع، لا لاحقة]
مُتَّخَذة	[Null Prefix, Noun, Objective Noun, Feminine, Single] [لا سابقة، اسم، اسم مفعول، مؤنَّث، مفرَد]

III. STRUCTURE OF THE PROPOSED DICTIONARY

The full-form words of the dictionary words required a very long time in searching for a given word if these words are stored in a single database. This simulate a big problem so the dictionary should be built in such a smart enough way that decrease the search time as much as possible to speed up the whole process of the dictionary.

To overcome these problems the idea is to distribute the full-form words into a large number of small size tables (files). A dictionary is proposed in this section in which the full form words are distributed over 30 tables.

A. *The proposed dictionary*

The database used for the proposed dictionary is designed to have 30 tables each of which holds all words that starts with the same alphabetic Arabic character. Table أ holds the words starts with the character أ , table ب holds the words starts with the character ب and so on. These are shown in figure 2. This way the whole data is distributed over thirty table rather than a single one.

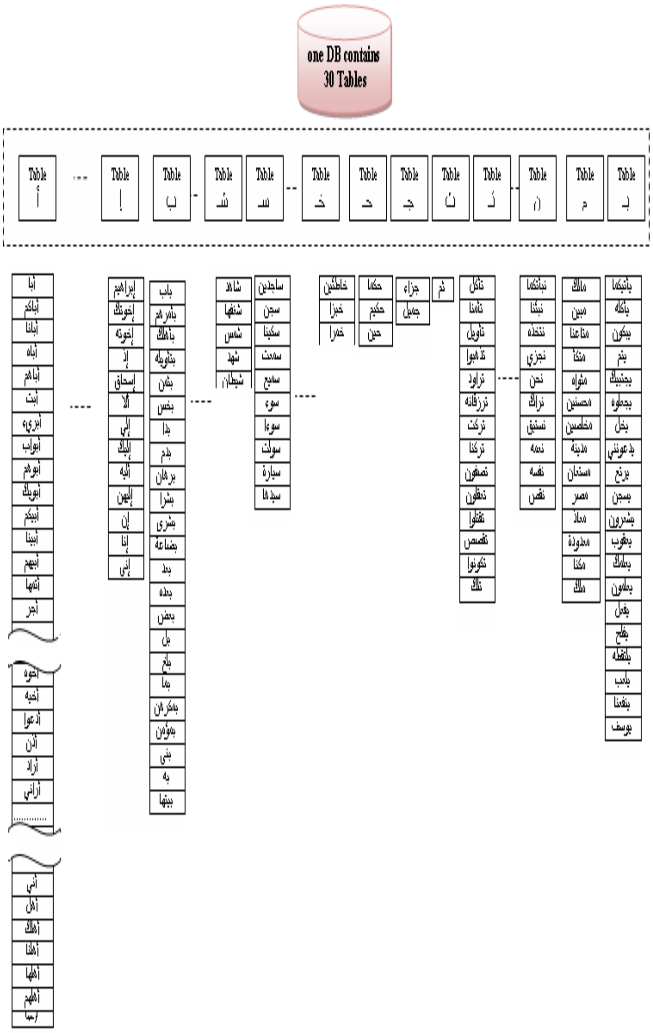
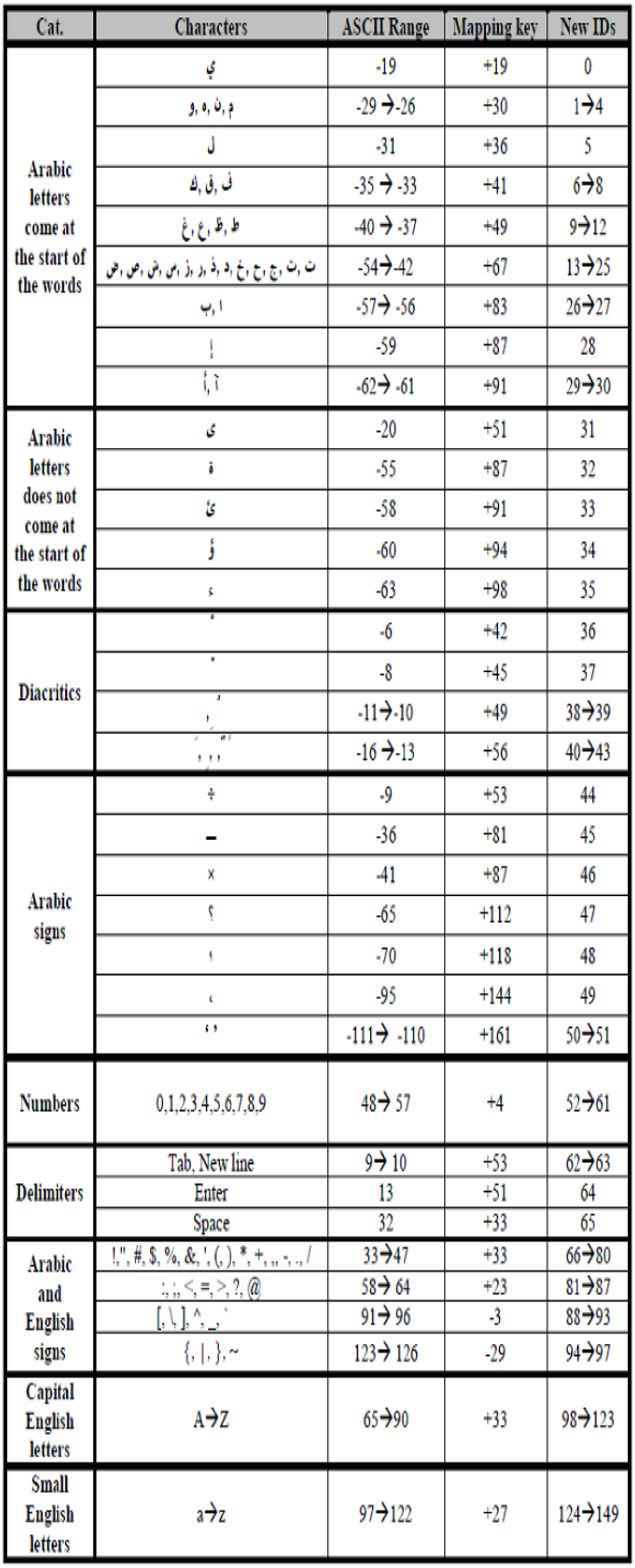


Fig.2: The design form of the database

The proposed dictionary requires the alphabets to have serialized IDs to increase the search speed, so all keyboard Arabic characters are collected and divided into groups, and then using the ASCII code of each character in the form of a tabulated function that maps the ASCII code to its ID. This can be shown in table 3. This table shows all Arabic characters distributed over nine groups with their ASCII1 codes, the mapping key, and the new IDs for all characters of the nine groups [11],[12]. This Table also shows 30 ASCII code of the alphabetic Arabic characters ,that starts the arabic words .The second group does not come at the starts of the words. This means that to map a certain character to its ID. we add the mapping key to the character’s ASCII code (e.g.: the ID of “أ” that has the ASCII code “-61” is “-61+91” = 30).

Table3. Characters mapping table



Cat.	Characters	ASCII Range	Mapping key	New IDs
Arabic letters come at the start of the words	ي	-19	+19	0
	م, ن, ه, و	-29 →-26	+30	1→4
	ل	-31	+36	5
	ف, ق, ك	-35 → -33	+41	6→8
	ط, ظ, ع, غ	-40 → -37	+49	9→12
	ت, ث, ج, ح, خ, د, ذ, ر, ز, س, ش, ص, ض	-54→-42	+67	13→25
	ا, ب	-57→ -56	+83	26→27
	إ	-59	+87	28
	آ, أ	-62→ -61	+91	29→30
Arabic letters does not come at the start of the words	ى	-20	+51	31
	ة	-55	+87	32
	ئ	-58	+91	33
	ؤ	-60	+94	34
	ء	-63	+98	35
Diacritics	ْ	-6	+42	36
	ّ	-8	+45	37
	ُ, ِ	-11→-10	+49	38→39
	ً, ٌ, ٍ, َ	-16 →-13	+56	40→43
Arabic signs	÷	-9	+53	44
	ـ	-36	+81	45
	×	-41	+87	46
	؟	-65	+112	47
	؛	-70	+118	48
	،	-95	+144	49
	‘ ’	-111→ -110	+161	50→51
Numbers	0,1,2,3,4,5,6,7,8,9	48→ 57	+4	52→61
Delimiters	Tab, New line	9→ 10	+53	62→63
	Enter	13	+51	64
	Space	32	+33	65
Arabic and English signs	!,", #, $, %, &, ', (,), *, +, ,, -, ., /	33→47	+33	66→80
	:, ;, <, =, >, ?, @	58→ 64	+23	81→87
	[, \,], ^, _, `	91→ 96	-3	88→93
	{, |, }, ~	123→ 126	-29	94→97
Capital English letters	A→Z	65→90	+33	98→123
Small English letters	a→z	97→122	+27	124→149

The dictionary structure is designed where each word is represented as follows:

Using the representation mentioned above for the words; the dictionary is implemented together with an example for the word " كتب" as shown in figure 3.

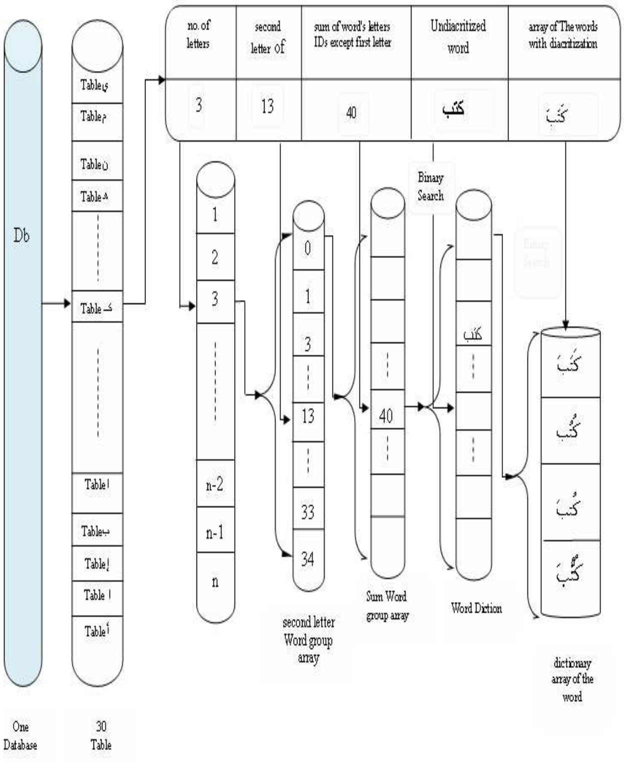


Fig3: shows how the word “كَتَبَ ” is found in the dictionary.

According to the structure shown in figure 3 the dictionary carries the word and its analyses (e.g. the word is “ كتب ” and the analyses are “ كَتَبَ , كُتُب , كُتِبَ , كُتُّبَ ”). So to find the ID of a certain word in the dictionary; the binary search is used once; to find the location of the Undiacritized together with the “Word analyses array” .The following equation is used to find any ID:

ID = D[n].Snd[i].S[l].W. FD (uw) (1)

Where:

1) D : is the “Dictionary” array.
2) Snd : is the “second Letter Word Group” array.
3) S : is the “Sum Word Group” array.
4) W : is the “Word Diction” array.
5) FD : is a function that apply the binary search technique on the “Word Diction” array to find an undiacritized word.
6) ID : is the location of the undiacritized word in the “Word Diction” array.
7) n : is the number of letters in the undiacritized word.
8) i : is the second letter ID.
9) l : is the summation of word's letters IDs except the first and second le in the undiacritized word letters
10) uw : is the undiacritized word string.
11) The “.” operator means that the right side of the “.” is an element of a structure (ex. A.B: this means that A is a structure and B is an element of A).
12) ex. to find the ID of the word “ كَتَبَ ”:

ID = D[3]. Snd [13].S[40].W. FD (“ كتب ”)

IV. EXPERIMENTAL RESULTS

In this section comparison studies are made between the results obtained from the proposed dictionary and those obtained from Mohammed attia in order to test the validity of the proposed dictionary. hundreds of words are used in the experiment the results of a sample of the words used in the search are given in table 4&5 and figure 4.

Table-4. Size of database against number of records

Database size Gigabyte (Gb)	number of record	Number of Record(ذهب)	Number of Record(شمس)	Number of Record(مدرسة)	Number of Record(جماعة)	Number of Record(كتب)
	Old dictionary	Proposed dictionary	Proposed dictionary	Proposed dictionary	Proposed dictionary	Proposed dictionary
9.16	134,217,728	3924810	3939810	4131810	4197810	4473924
13.4	209,715,200	6240480	6390480	6690480	6090480	69990506
18.8	268,435,456	7897830	8197830	8347830	8647830	8947848
31.4	400,000,000	11359320	12109320	12364320	12733320	13333333
69	536,870,972	14835690	15945690	16845690	17295690	17895699

Table5.size of database against execution time

Database size Gigabyte (Gb)	Execution time(ذهب)		Execution time(شمس)		Execution time(مدرسة)		Execution time(جماعة)		Execution time(كتب)	
	Old dictionary	proposed dictionary	Old dictionary	proposed dictionary	Old dictionary	proposed dictionary	Old dictionary	proposed dictionary	Old dictionary	proposed dictionary
9.16	59	0.400	59	0.490	59	0.530	59	0.560	59	0.600
13.4	60	0.65	60	0.71	60	0.82	60	0.84	60	0.9
18.8	90	1.095	90	1.11	90	1.15	90	1.26	90	1.3
31.4	120	1.470	120	1.690	120	1.720	120	1.850	120	2.000
69	240.7	2.120	240.7	2.199	240.7	2.210	240.7	2.250	240.7	2.259

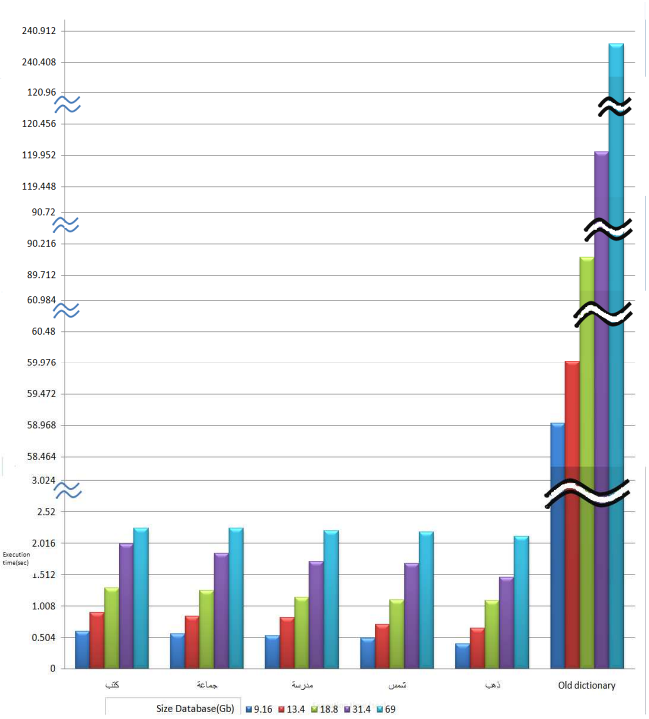


Fig4: Shows size of database against execution time.

V. CONCOLUSION

This paper introduced a technique suitable for distributing complex large scale database into a large number of small size ones. This way the time taken in searching in such database is minimized. These technique is suitable for use in the full form word Arabic dictionary.

From the results obtained from the comparison studies made in section 4 we can conclude the following

1) The time taken in the old dictionary almost constant and this time is increased as the size of the database increased (time taken to load the data).
2) The time taken in the proposed dictionary is much more less than that taken by the old one.

REFERENCES

[1] Otakar Smr, "Yet Another Intro to Arabic NLP",2005, http://ufal.mff.cuni.cz/~smrz/ANLP/anlp-lecture-notes.pdf

[2] Mohsen RASHWAN, Sherif ABDOU, Ahmed RAFEA" ,Stochastic Arabic Hybrid Diacritizer", IEEE trans. Natural Language Processing and Knowledge Engineering, pp.1-8, 24-27 Sept. 2009

[3] Mohamed Attia, Mohsen A. A. Rashwan, and Mohamed A. S. A. A. Al-Badrashiny, " Fassieh®, a Semi-Automatic Visual Interactive Tool for Morphological, PoS-Tags, Phonetic, and Semantic Annotation of Arabic Text Corpora", IEEE trans. AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 17, NO. 5, pp.916-925, JULY 2009

[4] http://Archimedes.fas.Harvard.edu/docs/Arabic. ", A Hybrid System for Automatic Arabic Diacritization", is found at http://www.rdi-eg.com ,pp.54-60.

[5] I. Zitouni, J. S. Sorensen, and R. Sarikaya, “Maximum entropy based restoration of Arabic diacritics,” in Proc. 21st Int. Conf. Comput. Linguist. and 44th Annual Meeting Assoc. for Comput. Linguist. (ACL); Workshop Comput. Approaches to Semitic Lang., Sydney, Australia, Jul. 2006 [Online]. Available: http://www.ACLweb.org/anthology/P/ P06/P06-1073

[6] Attia, M., Rashwan, M., Khallaaf, G., "A Formalism of Arabic Phonetic Grammar and Application on the Automatic Arabic Phonetic Transcription of Transliterated Words", NEMLAR int’l conference in Cairo, Sept. 2004. This paper is downloadable from http://www.RDI-eg.com/RDI/Technologies/paper.htm.

[7] Mansour Alghamdi, Zeeshan Muzafar ,"KACST Arabic Diacritizer",2012, http://www.mghamdi.com/KAD.pdf

[8] Mansour Alghamdi, Muhammad Khursheed, Mustafa Elshafei; Fayz Alhargan,Muhammed Alkanhal, Abu Aus Alshamsan, Saad Alqahtani, Syed Zeeshan Muzaffar, Yasser Altowim, Adnan Yusuf; Husni Almuhtasib , " Automatic Arabic Text Diacritizer", KACST, KFUPM,KSU, MODA, 18-6-2006

[9] KHALED SHAALAN,"Arabic Natural Language Processing: Challenges and Solutions", ACMTransactions on Asian Language Information Processing, Vol. 8, No. 4, Article 14, Pub. date: December 2009.

[10] Attia,"Automatic-Full-Phonetic-Transcription-of-Arabic-Script-with-and-without-Language-Factorization", http://www.rdieg.com/rdi/technologies/arabic_nlp.htm

[11] Mohsen A. A. Rashwan, Mohamed A. S. A. A. Al-Badrashiny, Mohamed Attia, Sherif M. Abdou, and Ahmed Rafea, "A Stochastic Arabic Diacritizer Based on a Hybrid of Factorized and Unfactorized Textual Features" IEEE trans. Auduio, speech, and language processing, vol.19, no.1, pp.166-175, junuary2011.

[12] M. Al-Badrashiny, “Automatic diacritization for Arabic texts,” M.Sc. thesis, Dept. of Electron. Elect. Commun., Faculty of Eng., Cairo Univ., Cairo, Egypt, Jun. 2009.

[13] M. Attia, M. Rashwan, and G. Khallaaf, “On stochastic models, statistical disambiguation, and applications on Arabic NLP problems,” in Proc. 3rd Conf. Lang. Eng.; CLE’2002, by Egypt. Soc. Lang. Eng. (ESoLE) [Online]. Available: www.ESoLE.org

[14] M. Attia and M. Rashwan, “A large-scale Arabic PoS tagger based on a compact Arabic PoS tags—Set, and application on the statistical inference of syntactic diacritics of Arabic text words,” in Proc. Arabic Lang. Technol. Resources Int. Conf.; NEMLAR, Cairo, Egypt, 2004.

[15] M. Attia, “A large-scale computational processor of the Arabic morphology, and applications,” M.Sc. thesis, Dept. of Comput. Eng., Faculty of Eng., Cairo Univ., Cairo, Egypt, 2000.

[16] Muhammad Atiyya ,Khalid Choukri ,Mustafa Yaseen (AU),"NEMLAR Specifications of the Arabic Written Corpus ", http://www.medar.info/The_Nemlar_Project/Publications/WC_design_final.pdf

[17] Indiana University,"Learning Arabic Morphology Using Statistical Constraint Satisfaction Models",April 2nd, 2005, http://www.casl.umd.edu/sites/default/files/rodrigues_ALS05_Learning-Arabic-Morphology-Using-Statistical-Constraint-Satisfaction-Models.pdf

[18] M. Attia, M. Rashwan, A. Ragheb, M. Al-Badrashiny, H. Al-Basoumy, and S. Abdou, "A Compact Arabic Lexical Semantics Language Resource Based on the Theory of Semantic Fields". Berlin, Heidelberg, Germany: Springer-Verlag, Aug. 2008, vol. 5221 [Online]. Available: www.SpringerOnline.com, Lecture Notes on Computer Science (LNCS): Advances in Natural Language Processing, LNCS/LNAI.

[19] M. Attia, “Theory and implementation of a large-scale Arabic phonetic transcriptor, and

applications," Ph.D. dissertation, Dept. of Electron. and Elect. Commun., Faculty of Eng., Cairo Univ., Cairo, Egypt, Sep. 2005.