Major Arabic Computational Linguistics Tools in Saudi Arabia: An Exploratory Study

Ruba Fahmi Bataineh
Professor of TESOL, Department of English, Prince Sultan University, Riyadh, Saudi Arabia
E-mail: rubab {at} yu.edu.jo

Rawan Abdulrahman Al-Wazzan
Teaching Assistant, Department of English, Prince Sultan University, Riyadh, Saudi Arabia

Abstract-**This paper surveys language applications from Saudi Arabia in particular in an attempt to provide an index and synopsis of available Arabic Computational Linguistics tools. The tools surveyed were chiefly developed by King Abdulaziz City for Science and Technology and King Fahad University of Petroleum and Minerals. They have exerted tremendous efforts in developing eight Arabic computational linguistics applications: KACST Arabic Diacritizer, KACST Arabic Phonetics Database, KFUPM's Experimental Arabic Text-To-Speech System, KACST Arabic Corpus, The Saudi Accented Arabic Voice Bank, the Automatic Stochastic Arabic Spelling Correction, the Arabic Morphological Analyzer and Arabic Stemmer, and the Arabic Phonetic Dictionaries for Speech Recognition. These efforts in the development and research of Arabic language tools are not only prominent but paramount in paving the way for future developments in the field of Arabic computational linguistics.**

Keywords--- Computational linguistics, language tools, language applications, natural language processing

I. INTRODUCTION

Computational Linguistics is an extremely multifarious field. However, despite the current liveliness of the discipline, with a plethora of associations and publications, it has not always been so. The beginnings of computational linguistics were extremely humble, most probably marked by warren Weaver's memorandum, back in 1949, that machine translation was in fact possible (Kay, 2003).

The field of computational linguistics has made significant breakthroughs in the past decade or so. With numerous applications, such as machine translation and speech recognition systems, easily available on the Internet and desktop computers, it has created a commercial enterprise that reaches millions of people around the globe. This development has been markedly facilitated by the advances in the theoretical representation and processing of language (Jurafsky & Martin, 2000).

The process of delivering information through natural language is an organized one, as can be clearly observed when attempting to converse with speakers of another language. The information communicated may seem clear to the speaker but is not necessarily understood by the listener if the language is not used properly. Computational Linguistics aims to reproduce the natural communication of information through technologically replicating the production and interpretation of language (Hausser, 2001). The present may be the best time to be working in the field of speech and language processing, with the Internet and the boom of web-based language tools as catalysts for application development (Jurafsky & Martin, 2000).

The Arabic language is one of the prominent six official languages in the world (UN Department for General Assembly and Conference Management, 2008) and the fifth most spoken with 280 million native speakers. Thus, there is a pressing need for a well-built language technology which can understand and serve these millions (Katz, Lin & Felshin, 2001). However, probably due to the complexity and distinctiveness of Arabic, computational linguistics tools are not as developed as those of other languages such as English (Abufardeh & Magel, 2008; Al-Daimi & Abdel-Amir, 1994; Habash & Rambow, 2005).

II. PROBLEM, SIGNIFICANCE AND PURPOSE OF THE STUDY

Although Arabic computational linguistics is just starting in the Arab world, a number of projects have been conducted for the purpose of developing intelligent Arabic language software. The present researchers have made an extensive search for research on projects catering to the Arabic language, but, unfortunately, the results were almost nonexistent. The research found is mostly dedicated to approaches to developing language tools. A number of articles did in fact discuss and explain, in great technical detail, a single computational linguistic tool, without ever synthesizing or addressing the aggregate context of computational linguistics tools in Arabic.

Thus, this research is a first attempt to identify and describe the existing number of Arabic language tools developed by Arab scholars. It surveys language applications from Saudi Arabia in particular in an attempt to provide an index and synopsis of available Arabic Computational Linguistics tools.

The tools surveyed herein were developed by King Abdulaziz City for Science and Technology (KACST) and King Fahad University of Petroleum and Minerals (KFUPM). The efforts of these institutions in the development and research of Arabic language tools are not only prominent, but paramount in paving the way for future developments in the field of Arabic computational linguistics.

III. THE TOOLS

In this section, the term 'tool' may refer to a language program that is being developed or one that is already published. Linguistic tools refer to projects that deal with technology and the human language. The purpose of these tools is to facilitate better communication between speakers of the same or different languages. In this section, the terms *tool*, *application* and *project* are used interchangeably.

The projects discussed below cater to different subfields of linguistics. More specifically, the Arabic Diacritizer, Phonetics Database, Text-to-Speech System, Voice Bank and Phonetic Dictionary may be most useful in the field of phonology whereas the Arabic Corpus may serve best in the field of corpus linguistics and Arabic Morphological Analyzer and Arabic Stemmer may be best used in the field of morphology.

A. KACST Arabic Diacritizer (KAD)

This tool is developed by King Abdulaziz City for Science and Technology for the purpose of diacritizing undiacritized texts.

The Arabic writing system is made up of 35 letters and 13 diacritics. Consonants are represented by letters whereas vowels are represented by diacritics. In current Arab media and other print and online publication, one seldom finds diacritized text. This may pose a strain on readers who would need to retrieve the diacritics, through a mental morphological, syntactic and semantic analysis of the text, to comprehend and avoid ambiguity. It is worth noting here that native speakers of Arabic are at a marked advantage over non-native speakers who are more likely to face difficulty in the comprehension of undiacritized Arabic texts (Alghamdi & Muzaffar, 2007).

Understanding the highly ambiguous Arabic terminology potentially poses a significant problem in many language tools. For example, the root 'كتب' [ktb] has several possible meanings, depending on its diacritics, as illustrated in Table 1 below.

Table 1 Illustrative Example

Undiacritized Word	Diacritized Word	English Transliteration	Meaning in English
كتب	كُتُبٌ	Kutub	books
كتب	كُتِبَ	Kutiba	was written
كتب	كَتَبَ	Kataba	he wrote

Most Arabic computational linguistic tools depend on the feature of diacritization to provide efficient language tools (Alghamdi & Muzaffar, 2007). In an Arabic text-to-speech system, many words are bound to be pronounced incorrectly if the text is undiacritized. Similarly, a search engine returned results potentially need diacritization to avoid ambiguity or misinterpretation.

The KASCT Arabic Diacritizer (henceforth, KAD) utilizes a two-step technique: generating a record of commonly used patterns of four diacritized Arabic letters called *quad-grams* and subsequently using this record to diacritize Arabic text.

In order to generate the quad-gram list, the KACST speech team developed a fully diacritized Arabic text corpus (henceforth, KDATD), comprised of a total of 231 text files containing an average of 1000 diacritized words each. The frequency and probability of each quad-gram extracted and calculated, and the ones with the highest frequency computed using the following formula:

$$P = \frac{Fh}{Ft}$$

Where *P* is the probability of a quad-gram sequence (such as س ع د), *Fh* is the frequency of the highest quad-gram sequence of letters and diacritics and *Ft* is the total of frequency of the quad-gram subsequence of letters with different diacritic combinations. The result was a database of 68,378 quad-grams which have the highest probability (Alghamdi & Muzaffar, 2007).

Two different sets of articles were used to test the accuracy of the system. The first set consisted of five undiacritized articles from KDATD. Not only was it fed into KAD for diacritization, but it was also done manually to compare machine and human diacritization, and the average error rate was 7.64%. As for the second tested set, which consisted of ten undiacritized articles from Al Riyadh

Newspaper, it was also done manually and using KAD. Its average error rate amounted to 8.87%. The average rate of both sets was 8.52% which is lower than any rate reported in the literature (Alghamdi & Muzaffar, 2007).

The Schematic Representation of KAD

Figure 1 below depicts the process of diacritizing the text fed into the tool.

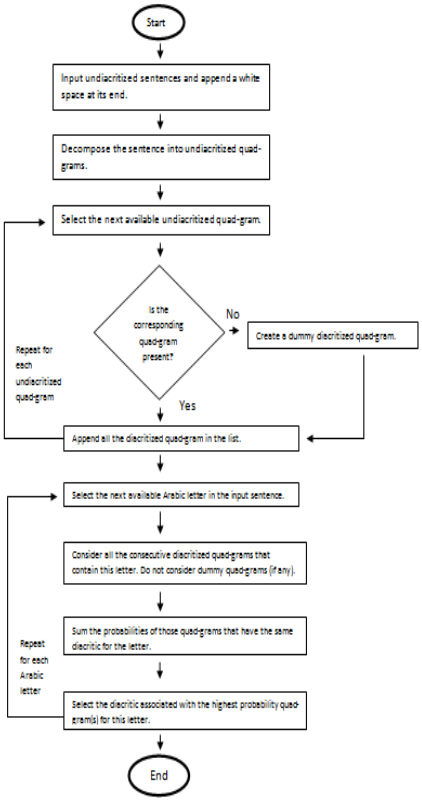

Figure 1: The process of converting Arabic text into allophones

***Source**: Automatic Arabic Text Diacritizer* http://www.ccse.kfupm.edu.sa/~elshafei/AG_publications.htm

Figure 1 above shows how the undiacritized text is fed into the system for diacritization. After processing the text and applying the appropriate quad- gram, probability is calculated. The diacritic with the highest probability is given to the corresponding letter until the alphabet of the whole text is completely diacritized.

B. KACST Arabic Phonetics Database (KAPD)

Like KAD, KAPD was developed by King Abdulaziz City for Science and Technology for the purpose of providing an Arabic phonetics database detailing the sound system of the Arabic language.

The Arabic phonetic system is believed to have been first written by Sibawayah in the eighth century in his renowned *Al-Kitaab* (Al-Nasser, 1993). Since then, a number of attempts have been made to contribute to the field of Arabic Phonetics. KASCT worked on developing an Arabic phonetics database, thereby facilitating easier research and more accessible Arabic sound data to interested researchers. KAPD culminated these efforts (Alghamdi, 2003).

The data included in KAPD exceeds 46,000 images, tables and audio files, the result of nine different experiments on Arabic native speakers with a Saudi accent. Images of the front and side views and voice recordings were taken, using a number of cameras and voice recorders. The tools used in the experiments to document the Arabic Phonetic system in the database included two Sony video cameras, video cassette recorder, Rhino-Laryngeal stroboscope, aerophone, palatometer, ElectroGlottoGraph, nasometer and a computerized speech laboratory (CSL) (Alghamdi, 2003).

The KAPD can be useed in a number of computational linguistics tools related to speech production. Speech-to-text and text-to-speech systems, speech therapy and forensic phonetics can all benefit from the data it can offer (Alghamdi, 2003).

C. The Saudi Accented Arabic Voice Bank (SAAVB)

Given the current state of Arabic speech databases, there is a growing need for much research and development. According to Alghamdi et al (2008), not only are the present databases underdeveloped, but they fail to cover dialectal diversity. Though Arabic is one of the most spoken languages in the world, its diverse dialects may in fact hinder the development of an all encompassing Arabic speech database. Two prominent institutions, KACST and IBM Egypt, collaborated to develop a speech database specific to the different Arabic native dialects present in Saudi Arabia. The project was called the Saudi Accented Arabic Voice Bank (SAAVB). This marked the first attempt to be made at developing a comprehensive database that covers the many dialects found in the country (Alghamdi et al, 2008).

To design the voice bank, the team divided the data collection process into four major steps: prompt sheet design, speaker selection, speech recording and transcription. A total of 1033 speakers took part in the project. The speaker's gender, age, city and region were recorded. Every speaker was given unique prompt sheets, each consisting of 59 items. 83% of the items were written in Modern Standard Arabic and 17%

elicited spontaneous responses. Though each speaker received a unique sheet, two specific sentences appeared across all the speakers' sheets. The purpose of this was to identify dialectic variations among participants. The KACST and IBM Egypt project collaboration culminated in the Saudi Accented Arabic Voice Bank, which has been used to develop and train Arabic speech recognition systems. It may also be available to developers and researchers -after signing a contract with KACST- for the purpose of research or product development. SAAVB is also potentially beneficial to speaker verification and language identification applications.

D. Arabic Phonetic Dictionaries for Speech Recognition

Ali et al (2009) suggests that a phonetic or pronunciation dictionary is an important tool in a speech recognition system. The dictionary includes the pronunciations of words in the language and serves as a mediator between the language and acoustic models of the speech recognition system. This tool depends on a large set of phonetic rules for its development.

E. KFUPM's Experimental Arabic Text-To-Speech System (KFUPM's ATTS)

This system, developed by King Fahd University of Petroleum and Minerals for the purpose of converting written Arabic text into spoken language, focuses on segmenting Arabic text to allophones. The total number of 150 allophones were used, which were each assigned to its counterpart in the International Phonetic Alphabet (IPA). Table 2 below illustrates the process of converting Arabic text into allophones:

Table 2: The Process of Converting Arabic Text into Allophones

Arabic Text	*Arabic Allophone*	*IPA Allophone*
بسم الله الرحمن الرحيم	ب س م ا ل ل ه ا ر ح م ن ا ر ح ي م @	/BB2/ /EH/ /SS/ /PA2/ /MM/ /IH/ /IR1/ /PA1/ /LL/ /AT_L/ /PA2/ /HH/ /IR1/ /PA1/ /RR3/ /AA/ /H3/ /PA2/ /MM/ /AE:/ /PA1/ /NN1/ /IR1/ /PA1/ /RR3/ /AA/ /PA2/ /I1/ /IH:/ /MM/ /PA5/.

After the text is segmented into allophones, these allophones are digitally encoded per the rules of phonology, such as lexicon lookup, breathing, prosodic, pronunciation, and synthesis model rules. KFUPM's Experimental ATTS system only considers phonological rules with no regard to the morphological attributes of the text (Elshafei, 1991).

Schematic Drawing of a TTS System:

Figure 2 below provides a representation of the process of the conversion of Arabic text to speech.

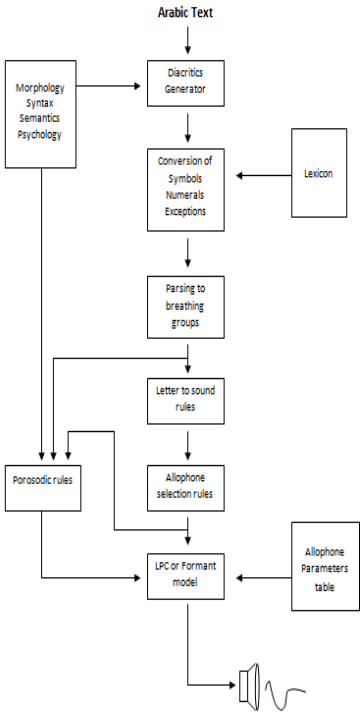

Figure 2: Schematic drawing of the text-to-speech system (*Source:* Elshafei (1991)).

After the text is fed into the system, diacritics are assigned and phonological rules applied to produce spoken output.

F. Automatic Stochastic Arabic Spelling Correction with Emphasis on Space Insertions and Deletions

Misspelled text often results in ambiguity, misunderstanding, and miscommunication. For the longest time, manual editing and detection of misspelled words have been used to alleviate these problems. However, these strategies are time-consuming and, in an age where every second counts, the need for developing a dependable spelling correction system is dire.

As they stand now, Arabic spelling checkers lag behind their English counterparts. They mostly fail to account for context (Alkanhal et al, 2012).

Even though an actual tool has not yet been developed to compete with current spelling check systems, a specialized team from KASCT has designed a stochastic approach for automatic correction of spelling errors. This approach takes into consideration all the types of spelling errors and,

therefore, potentially outperforms current Arabic spelling check systems.

Assigning the correct form of a misspelled word is done in three phases: detecting the error, generating a candidate, and selecting the most appropriate candidate (Alkanhal et al, 2012),. In the error detection phase, the type of error must be identified. Table 3, below, shows the six major kinds of spelling errors:

Table 3: Types of Spelling Errors

Type of Error	Explanation	Example
(a) Substitution	The substitution of letters produces a correctly spelled yet different word.	Misspelled Word: يشرب Correct Word: يتسرب
(b) Insertion	Space is inserted in a correctly spelled word producing a correctly spelled phrase.	Misspelled Word: مت طور Correct Word: متطور
(c) Deletion and substitution	Letters are substituted while others are deleted. Thereby, producing a word that is not part of the language's dictionary.	Misspelled Word: بتبذا Correct Word: يكتبها
(d) Deletion or insertion	Space is either inserted or deleted from a word.	Misspelled Word: بريقالذهب Correct Word: برنق الذهب
(e) Deletion, substitution and insertion	Space is inserted after some letters are substituted and others deleted resulting in a non-word.	Misspelled Word: ت بتا Correct Word: يكتبها
(f) Substitution and insertion	Space is inserted after some letters are substituted resulting in a correctly spelled word or phrase.	Misspelled Word: سكت بها Correct Word: يكتبها

Types a, b, and f, in Table 3, are known as Semantic Hidden Errors (Zribi, Mejri & Ahmed, 2007).

N-gram algorithms, morphological analysis, and dictionary lookup are techniques that can be used to determine whether or not a word is correctly spelled. The stochastic spelling correction approach uses the dictionary lookup technique (Raynaert, 2011).

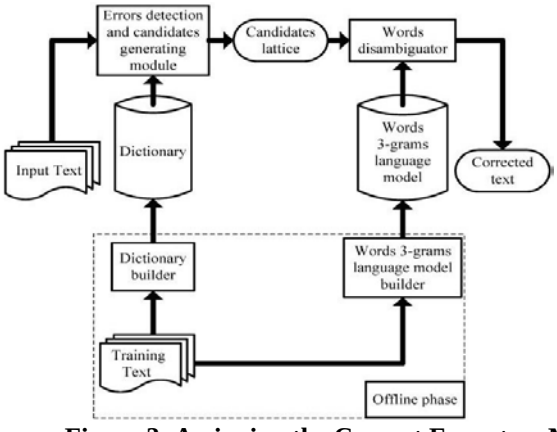

Figure 3: Assigning the Correct Form to a Misspelled Word (Source: Alkanhal et al (2012))

In the second phase, candidate generation, the objective is to find appropriate corrections for the misspelled word. Levenshtein distance (Levenshtein, 1966), which involves calculating the minimum number of insertion, deletion, substitution or transportation edits required to correct a misspelled word, is a well-known method for candidate retrieval (Shalaan, Aref & Fahmy, 2010).

The final phase in the process of assigning the correct form to a misspelled word is selecting the best candidate from a list of retrieved candidates. Figure 3 illustrates the process of assigning a misspelled word its correct form.

G. Arabic Morphological Analyzer and Arabic Stemmer

Most computational linguistic tools need a morphological analyzer to break down text into its basic components, which has driven the development of a morphological analyzer and Arabic stemmer. According to Alghamdi et al (2009), a team of professionals from KACST has developed an algorithm-based Arabic morphological analyzer which analyzes Arabic vocabulary depending on its morphological and grammatical properties. A large linguistic corpus is needed in this project to provide these properties. To support this system, KACST has also built an Arabic Stemmer. This tool is used to strip words of their suffixes, prefixes and infixes to extract the stem word. An Arabic Morphological Analyzer comes very handy not only in machine translation but also in information retrieval and text analysis and generation.

H. KACST Arabic Corpus (KAC)

KAC was jointly developed by King Abdulaziz City for Science and Technology and King Abdullah's Arabic Content Initiative for the purpose of providing a linguistic corpus for the Arabic language.

No published literature is available to fully examine the KAC linguistic tool. However, it could be gleaned form the KAC website.= that the project is still in its early stages. The corpus includes a total of 732,780,509 Arabic words from 869,800 different texts collected from over 20 Arab countries (mainly from Saudi Arabia, Lebanon, Egypt, Kuwait, Iraq and Syria). Even though the corpus has texts which date as far back as 800 A.D., most texts were written after the year 2000. Retrieved text mostly came from newspapers, magazines, manuscripts and books. KACST Arabic Corpus is available online at http://www.kacstac.org.sa/ and can be used by the general public.

IV. DISSCUSSION

This paper showcases eight language applications, seven of which are developed by KACST and one by KFUPM. The absence of diacritics in current Arab media motivated the development of KAD. The need for a phonetics database to record the Arabic sound system was behind developing KAPD, SAAVB and phonetic dictionary. Similarly, ATTS was

developed to convert written Arabic text into spoken language. The need to breakdown words to even smaller components prompted the development of the Arabic morphological analyzer and stemmer. Similarly, the Arabic corpus as a reference motivated the development of KAC, and the need for a reliable spelling check system prompted the development of the stochastic approach.

One cannot ignore the intricate ties between the various sub-fields of linguistics. Hence, even though some tools may fall under one field, they will most certainly be useful for another. Oftentimes, for language tools to perform efficiently, they sport interdependence. For example, an application like KASCT's Arabic Diacritizer (KAD) needs a well-built corpus like KASCT's Arabic Corpus (KAC) to perform diacritization. Likewise, KFUPM's ATTS would need a tool like KAD to perform its first step of converting text into speech, which is assigning diacritics to the input. It will also use the data KASCT's Arabic Phonetic Database (KAPD), Saudi Accented Arabic Voice Bank (SAAVB), and Phonetic Dictionary have a lot to offer in terms of sound and pronunciation. Machine Translation and Information Retrieval tools will benefit greatly from the Arabic Morphological Analyzer and Arabic Stemmer to extract word stems and yield better results. They will also use the Automatic Stochastic Arabic Spelling Correction Approach to correct misspelled text.

The aforementioned tools serve as building blocks for the foundations of Arabic computational linguistics applications. This study does not presume claim that the tools surveyed above are the only ones available in Arabic. More projects in the field are either in existence or under development. More work worldwide is being dedicated to supporting the Arabic language in the field of computational linguistics and natural language processing. In Saudi Arabia in particular, these eight projects are not only the most prominent, but also the most accessible.

V. CONCLUSION

It may be deduced that the field of computational linguistics is relatively young compared to other areas of study and particularly more so in Arabic. With the sole purpose of bettering machine language production and interpretation, computational linguistics found its footing with the advent of technology. Understandably, the advances of Arabic computational linguistics are still in the early stages. In Saudi Arabia, two institutions, namely King Abdulaziz for Science and Technology and King Fahd University of Petroleum and Minerals, have dedicated marked efforts to developing Arabic computational linguistics tools.

VI. RECOMMENDATIONS

After reviewing the most prominent computational linguistic projects in Saudi Arabia, it is fairly obvious that further development is needed for these tools to be readily accessible to professional linguists and programmers. Moreover these tools may be incorporated in undergraduate or graduate programs in both linguistics and computer science as examples of the Arabic computational linguistic applications available.

REFERENCES

1. Abufardeh, S. & Magel, K. (2008). Software Localization: The Challenging Aspects of Arabic to the Localization Process (Arabization). IASTED Proceeding of the Software Engineering SE 2008 (275-279), Innsbruck, Austria. Retrieved 25 December 2012 from http://www.cs.ndsu.nodak.edu/~abufarde/research.html.

2. Akour M., Abufardeh S., Magel K. & Al-Radaideh Q. (2011). QArabPro: A Rule Based Question Answering System for Reading Comprehension Tests in Arabic. *American Journal of Applied Science*. Retrieved 25 December 2012 from http://www.cs.ndsu.nodak.edu/~abufarde/research.html.

3. Al-Daimi, K. & Abdel-Amir, M. (1994). The Syntactic Analysis of Arabic by Machine. *Computers and the Humanities*, 28(1), 29-37. Retrieved 18 March 2013 from http://www.jstor.org/discover/10.2307/30200308?uid=3738952&uid=2&uid=4&sid=21102305256341.

4. Alghamdi M., Alhargan F., Alkanhal M., Alkhairy A., Eldesouki M., & Alenazi A. (2008). Saudi Accented Arabic Voice Bank. *Journal of King Saud University-Computer and Information Sciences* 20, 45-64.

5. Alghamdi M., Mokbel C. & Mrayati M. (2009). Arabic Language Resources and Tools for Speech and Natural Language: KACST and Balamand. *2nd International Conference on Arabic Language Resources and Tools.* Cairo, Egypt. 22-23 April.

6. Alghamdi, M. & Muzaffar, Z. (2007). KACST Arabic Diacritizer. *The First International Symposium on Computers and Arabic Language*, 25-28 March.

7. Alghamdi, M. (2003). KACST Arabic Phonetics Database, *Proceedings of the Fifth International Congress of Phonetics Science (*3109-3112), Barcelona,.

8. Ali M., Elshafei M., Alghamdi M., Almuhtaseb H., & Alnajjar A. (2009). Arabic Phonetic Dictionaries for

Speech Recognition. *Journal of Information Technology Research*. 2(4). 57-80.

9. Alkanhal M., Al-Badrashiny M., Alghamdi M., & Al-Qabbany A. (2012). Automatic Stochastic Arabic Spelling Correction with Emphasis on Space Insertions and Deletions IEEE Transactions on Audio, Speech, and Language Processing, 20(7), 2111 - 212

10. Al-Nasser, A. (1993). Sibawayh the Phonologist: A Critical Study of the Phonetic and Phonological Theory of Sibawayh as Presented in His Treatise Al-Kitab, Kegan Paul Intl.

11. Elshafei, M. (1991). Toward an Arabic Text-to-Speech System. *Arabian Journal of Science and Engineering*, 16(4B), 565-583, Retrieved 27 March 2013 from http://www.ccse.kfupm.edu.sa/~elshafei/MES_AJST91.pdf.

12. Habash, N. & Rambow, O. (2005). Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. Proceeding of the 43rd Annual Meeting on Association for Computational Linguistics, 573-580, June 25-30, Ann Arbor, Michigan. Retrieved 5 June 2013 from http://www1.cs.columbia.edu/~rambow/papers/habash-rambow-2005a.pdf.

13. Hausser, R. (2001). *Introduction. Foundations of Computational Linguistics: Human-Computer Communication in Natural Language*. New York: Springer.

14. Jurafsky, D. & Martin, J. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River (New Jersey): Prentice-Hall, Inc.

15. Katz, B. Lin, J. & Felshin, S. (2001). Gathering Knowledge for a Question Answering System from Heterogeneous Information Sources. *Proceedings of the Workshop on Human Language Technology*, ACL-2001, Toulouse. Retrieved 5 June 2013 from http://groups.csail.mit.edu/infolab/publications/Katz-etal-ACL01.pdf

16. Kay, M. (2003). Introduction. In M. Ruslan, *The Oxford Handbook of Computational Linguistics* (xvii-xx). New York: Oxford University Press Inc.

17. Kukich K. (1992). Techniques for Automatically Correcting Words in Text *ACM Computing Survey*, 24(4), 377–439.

18. Levenshtein V. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals *Soviet Physics Doklady*, 10(8), 707-710.

19. Reynaert M. (2011). Character Confusion Versus Focus Word-Based Correction of Spelling and OCR Variants in Corpora *in International Journal on Document Analysis and Recognition* , 14(2),1-8.

20. Shaalan K., Aref R., & Fahmy A., (2010). An Approach for Analyzing and Correcting Spelling Errors for Non-native Arabic Learners. *Proceedings of the 7th International Conference on Information Systems. INFOS2010*, the special track on Natural Language Processing and Knowledge Mining, 28-30 March, Cairo, Egypt.

21. UN Department for General Assembly and Conference Management, (2008). UN Official Languages. Retrieved 4 February 2013 from http://www.un.org/Depts/DGACM/index.shtml.

22. Zribi C., Mejri H., & Ahmed M. (2007). Combining Methods for Detecting and Correcting Semantic Hidden Errors in Arabic Texts. *Proceedings of the 8th International Conference on Computational Linguistics Intelligent Text Processing.*

AUTHOR PROFILES

Ruba F. Bataineh, PhD is currently a professor of TESOL (Applied Linguistics) in the Department of English at Prince Sultan University (Formerly Yarmouk University, Jordan). She has published over three dozen articles in renowned international and regional scholarly journals.

Rawan A. Al-Wazzan got her BSc in Computational Linguistics from Prince Sultan University in 2012. She is currently a Teaching Assistant in the Department of English at Prince Sultan University.