Information Retrieval of Text with Diacritics

Khalid Saleh Rabeh Aloufi

Department of Computer Science, College of Computer Science and Engineering, Taibah University, Madina, KSA,

Summary
Information retrieval systems are mostly engineered, optimized and designed for English language. However, every Language has some special or common features. Diacritics are one of the special features of some non English languages that are not fully supported. Non English Internet users are not easily searching for a great range of information. The proposed Information retrieval model specifically designed for a text with diacritics. The proposed model is implemented to retrieve information from database and shows accurate results. The proposed system can be integrated in search engines and desktop text editors and to any language featured with diacritics, such as Arabic, Greek, Hebrew, and Korean.

Key words:
Information Retrievals; Natural Language Processing; Regular Expressions; Search Engines.

1. Introduction

Information retrieval systems are important tool to find the required information easily and in a short time. Information retrieval is defined as "locating quantities of data stored in a database and producing information from the data" [1]. Search engine is defined as "software that performs the search on a database or title" [2]. The information retrieval systems can be developed as a search engine to facilitate looking for the required information in a document.

The population of online non English speakers is Over 60% of the total population [3]. The search results are not highly successful of non English queries [4].

Every language has its features that are required to develop natural language processing systems. For instance, examples of English language features are the capitals and small letters. Arabic, Greek, Hebrew, Korean and Sanskrit are example languages that feature the use of diacritics. Diacritics are used to indicate the pronunciation of letters or the sounds. One of the main Arabic diacritics is listed in table 1.

Arabic language uses short vowels written as diacritic with every letter. Different diacritics with a word of the exact match of letters will result in different pronunciation and meaning.

Diacritics are written below or above the letter. However, for a file of Unicode format it is saved sequentially after the letter. Unicode uses two byte for each character in UTF-16 and a mixture of one and two byte for each character in UTF-8.

Searching text is a basic tool in all text editors and web search engines. It is simpler for the user to not enter diacritics when writing or searching in web sites. In fact, it is common assumption that the user will not enter the diacritics of the words either to write or to search for information.

For search engines development, it is common practice for Information Retrieval systems to remove diacritics from documents [5] [4] [6]. Then, document becomes available for the users and computer systems to search for information.

However, removing diacritics from documents that comes originally with diacritics will remove some values of the document such as understanding the meaning of the words because in some languages such as Arabic, diacritics are useful for understanding the meaning. The document that usually comes with diacritics is the religious and history books as well as children stories. Also education book usually consider diacritics. This means the set of information with diacritics is not small if not even large and considered a main interest for non English search engines users.

For IR system development, the system is either has to remove diacritics from the documents and then return the results without diacritics or with diacritics using complex system. The second option is to develop a system that search text with diacritics without changing the sources.

This study suggests not removing diacritics from the document or any information repository but changing the user input to be with diacritics. This is the main aim of this research. Also, the number of returned result increases as will be shown later.

Boolean retrieval is one of the models of information retrieval [2]. It is easy to implement. However, it includes

Manuscript received August 5, 2010
Manuscript revised August 20, 2010

some limitations such as equal terms weights, no ranking for results. On the other hand, Vector space model (VSM) is one of the main methodologies to search for data [7] [8] [9].

VSM is advanced model and solve the mentioned limitations of the Boolean model. The proposed model is Boolean model with some advanced features to search a text with diacritics. The model also referred to as ranked model of VSM for Information Retrieval.

The proposed model is novel and can be used to develop a tool in text editors as well as search engines. Also, it can be applied in extended Boolean model or VSM model to enhance retrieving information.

Different character classes are suggested to optimize the regular expressions for different encoding such as Unicode [10]. Search engines and sequential query language (SQL) is designed with respect to the Latin encoding. It does not fully support Unicode compared to the support of the Latin encoding [10] [3]. The effectiveness of search engines is expected to increase if the specific features of individual languages are considered [3]. The search engines should detect the language from the input of the users and response accordingly.

Table 1 Main Arabic Diacritic

Unicode	Arabic diacritics meaning	Pattern
064B	ARABIC FATHATAN	ً
04C	ARABIC DAMMATAN	ٌ
064D	ARABIC KASRATAN	ٍ
064E	ARABIC FATHA	َ
064F	ARABIC DAMMA	ُ
0650	ARABIC KASRA	ِ
0651	ARABIC SHADDA	ّ
0652	ARABIC SUKUN	ْ
0653	ARABIC MADDAH ABOVE	آ
0654	ARABIC HAMZA ABOVE	ٱ
0655	ARABIC HAMZA BELOW	إ

The regular expressions are provided functions by most DBMS [10]. However, it is not fully compatible with Unicode [10]. Supporting Unicode is a support for all spoken languages. The compatibility is partial as tested in this study and will be presented in the experiments section.

There is provided support for Unicode in different DBMS, but the design and optimization is based for Latin encoding. Some functions are not performing as expected as detailed next.

The most popular pattern matching operator in SQL and other popular database languages is LIKE which supports wildcard characters for matching a single character or a sequence of characters. One of the solutions for such problem is providing new classes definitions in the basic SQL and implemented and supported by any DBMS.

Proposed classes are for diacritics are as follows. [:dia:] is proposed to match any diacritic symbol, or a set of diacritic marks. [:acc:] is proposed to match an accent diacritic. [:dvow:] is proposed to match a vowel diacritic. [:tone:] is proposed to match a diacritic representing a tone. “x DILIKE y “ is proposed pattern to not include diacritics in matching of regular expressions [10].

The above features are suggested improvement for SQL. However, if not available, system developer has either to self develop the required functions, otherwise, the programming languages that interface with SQL have to find a way to interface with any query not supported by SQL.

The Diacritics are treated as characters. However, multi-byte, such as Unicode, is incompatible with some SQL operators. For instance, the MySQL operators RLIKE and REGEXP are not compatible with multi-byte characters [11].

The PHP web programming language has includes a set of substituting functions for multi-byte characters encoding such as Unicode.

The scope of this study is to develop a search tool that is useful for text with diacritic. The tool is applied on an Arabic text. The system is developed based on the Arabic language features. The system can be developed for any other language that features diacritics.

In general, the whole set of features of the target language must be considered during the development of the systems.

In this study, diacritics are not extracted from the original document as the case of systems developed to search Arabic document [12] [6]. Searching a document with its diacritic is not easily managed [6]. This study develops a system that successfully overcomes this challenge.

The paper is organized as follows. The next section describes the System Model, which consists of the

following stages: input processing, input terms, searching and search results. Results and Analysis is the section for discussing the results. Development and research suggestions are summarized in the section Conclusion and Future work. This paper ends with the references used in this research.

2. The System Model

The system consists of four processes, which are "input processing", "input terms', "searching" and "search results" as shown in figure 1. Input processing is the process of defining the characters of the entered term. Input terms are the step of the definition of all the possible terms the system should search for. Searching is the process of a sequences of SQL query to search for all the possible terms. A search result, the last process, is the result set that returned to the user.

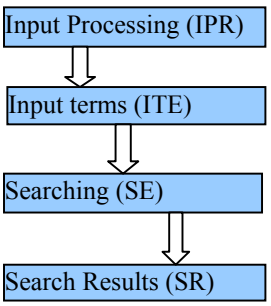

Figure 1 the model processes

2.1 Input processing

Input processing (IPR) is the first process. IPR define the characters of the entered word. Each letter is followed by another letter until the end of the word. This step should have included user input validation. For instance, any number or diacritics considered as error and ask the user to be removed. In the future the model can be developed to consider numbers and diacritics.

2.2 Input terms

Input Terms (ITE) is the second process. ITE is the step of the definition of all the possible words derived from the original word entered by the user. The possible words contain the same letters of the original word with exact sequence.

However, letters will be followed by either a diacritics or by the next letter in the word. If the letter is followed by a letter, it means the diacritic is not defined for this letter. If a diacritic is defined for a letter, it is placed after the letter. In the file format, the diacritic symbol sequentially follows the letter.

The result will be a result set of possible words. The search will include the original words and the generated words as well.

The user is searching for a word. Then, the system will generate all the possible sequences of the word with different diacritic. For example the word قال, this is pronounced in English as 'GAL'. When the user enters the word, this step generates the word with all possible diacritics.
As for the example of the word قال or 'GAL', there are different possible generated words. For example, possible generated words are قَال or قَلَ or قَلْ and the list extends to all the possible words.

In general, if " x_n" is a letter and "-" is a diacritic; where *n* is the order of the letter in the word. If a word of three letters, then the original word will be "$x_1x_2x_3$".

The generated terms will be "x_1-x_2-x_3-", where the dash "-" will be replaced by one of the diacritics or nothing.

Nothing means the letter is flowed by next letter in the word. If x_1 is followed by no diacritic, but x_2 and x_3 are followed by diacritics, then the word will be like "x_1x_2-x_3-".
If the word is "$x_1x_2x_3$", then the possible generated words will be "$x_1x_2x_3$"," x_1-x_2x_3 "," x_1x_2-x_3 " and "x_1-x_2-x_3". Each of these terms may have results and may not.

For each diacritic in Arabic, there are eleven possible diacritics, which are listed in table 1. For example, if a word is composed of one or two or three letters, then it will be 11 or 121 or 1331 words, consequently.

The performance improves using regular expression, with which the numbers change to 2, 4, 8, 1024 for words of one, two, three or ten letters, consequently.

In general, the number of generated words is 2^{n-1}, where *n* is the number of letters of the original word.

As the number of letters increase the number of possible pattern increase. The time required to get results increase as the number of the number of letters increases. The algorithm of this step is detailed in figure 2.

Variables:

Original Word: entered by user

Words Array: contains words that will be forwarded to the search engine.

The algorithm works as follows:

- *User enters the Original Word.*
- *Generate all possible Words*
- *Add Words to the Words Array*

Figure 2 ITE Algorithm

2.3 Searching

Searching (SE) is the third process. The process is a search engine to process the words from ITE. SE is invoked by sequences of queries of all the words using the regular expressions.

The query replaces any dash "-" with any characters. The main character concerned is diacritics. This will solve the problem of different matches of different diacritics. The query should return a number of results either zero or more.

2.4 Search result

A search result (SR) is the last and fourth stage. SR is the result set that returned to the user. It will include all the matches of the words generated by the ITE. As mentioned in the algorithm in figure 2, there will be different generated words.

3. Results and Analysis

The model is applied on Arabic Language. The user will not enter any diacritic. The system searches for only one word at a time for each search. The system will consider all possible diacritics after each letter.

The system is developed using JAVA programming language and a MySQL database of Arabic text. Each paragraph of the e-book is saved in a record. This e-book contains more than 2000 sections and therefore more than 2000 records in the database.

Arabic Unicode letters Ranges from 0600 to 06FF in the Hex reference of the Unicode standardization [13]. The Arabic letters in Unicode range from 0621 to 063A and from 0641 to 064A [13]. The Arabic diacritics range from 0618 to 061A and from 064D to 0656[13].

The Arabic diacritics will be considered in SE but as mentioned early that the system assumes the user will search for a word with letters only. There are 36 number of Arabic letter and 11 numbers of diacritics.

Table 2 shows the result of searching for the word "GAL"," قال". There is 44, 2022, 0 and 7 search results for the word "GAL"," G-AL"," GA-L" and "G-A-L" consequently. The system is successful in returning results for the different words generated by ITE.

Table 2 Generated words and the results query of each word using regular expression

Generated words expressions , English	**Generated words expressions, Arabic**	**Number of results of the query**
GAL	قال	44
G-AL	ق-ال	2022
GA-L	قا-ل	0
G-A-L	ق-ا-ل	7

4. Conclusion and Future work

The paper proposes a novel model that can be used to search for any diacritics words in databases or web pages using search engines. The proposed system is developed because of the limitations of the support found for non Latin languages in the Internet.

The proposed system has some limitations, such as the performance gain because of the number of queries generated. The model need more advanced algorithm to increase performance. The model can be integrated in search engines mainly and text editors Find property found in several text editors.

The system generates a sequence of queries for words. Some words cannot be found in because the combination of diacritics does not exist. Some combination of letters may not exist in the language or the sample document used.

The proposed model consists of four steps, which are "input processing", "input terms', "searching" and "search results". The system starts by getting the word from the user, then processing the character of the words in IPR, followed by the definition of all possible words in ITE, and finally searching for the words in the SE. The model ends up with a set of returned results in SR.

Future studies will include the addition of VSM model to the system. However, this will decrease the number of words that will be searched for because words with zero results will not included in the search. Performance issues will meet challenges and will need advance information retrieval system to minimize words with no weight or no records at all.

References

[1] S.M.H. Smith, *Dictionary of Information Technology* (Peter Collin Publishing, 2002).

[2] Lancaster, F. Wilfrid and Fayen, Emily Gallup, *Information Retrieval (*Melville Pub. Co., Los Angeles, 1973).

[3] Fotis Lazarinis, Jesus Vilares Ferro, John Tait, Improving Non-English Web Searching (iNEWS07), *SIGIR Forum* 41(2) (2007) 72-76

[4] S. Beitzel, U. Syed, E. Jensen, O. Frieder and D. Grossman, "On the Development of Name Search Techniques for Arabic", *Journal of the American Society for Information Science and Technology*, 2006, 57(6), pp.728– 739.

[5] A. Alhajjar, Mohammad Hajjar, Khaldoun Zreik, Classification of Arabic Information Extraction methods, *2nd International Conference on Arabic Language Resources and Tools*, April 2009, Cairo, Egypt.

[6] Bassam Hammo, Mahmoud EL-Haj, Azzam Sleit, Enhancing Retrieval Effectiveness of Diacritisized Arabic Passages Using Stemmer and Thesaurus: *The 19th Midwest Artificial Intelligence and Cognitive Science Conference* (Cincinnati, OH, USA, 2008).

[7] G. Salton, A. Wong, and C. S. Yang, A Vector Space Model for Automatic Indexing *Communications of the ACM* 18 (11) (1975)613–620.

[8] Salton, G., and McGill, M.J., *Introduction to Modern n information Retrieval* (McGraw-Hill Book Company, New York, 1983).

[9] Salton, G., Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer, (Addison Wesley, MA, 1989).

[10] Sudeshna Sarkar, Regular Expression Matching for Multi-script Databases, *Bulletin on the Technical Committee on Data Engineering* 30(1) (2007)17-29.

[11] Sun Microsystems, MySQL 5.5 Reference Manual (2010). Available at: www.mysql.com (accessed 16 Mar 2010)

[12] N. Thabet, Stemming the Qur'an, *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, 2004.

[13] The Unicode Consortium, *Unicode Standard. Version 5.0*, (Addison-Wesley, 5th edition, November 2006).

Aloufi Khalid received the B.S. degree from King Fahd University of Petroleum and Minerals, Saudi Arabia, and MSc and PhD degrees from Bradford University, UK, in Informatics, in 2006. During 2002-2006, he stayed in Networks and Performance Engineering Research Group and Laboratory, Computing, Bradford University. Currently he is an Assistant Professor. He is the Vice Dean of the college of Computer Science and Engineering and the Head of Computing Information Systems department at Taibah University, Saudi.