New Searching Rules to Find Variant Names in Arabic

M.M. Badr[1], A.A. El-Harby[1] and A.M. Riad[2]

[1] Department of Mathematic, Faculty of Science, Mansoura University, New Damietta, Egypt.
[2] Department of Information System, Faculty of Computers and Information Systems, Mansoura University, Egypt.
mohfbadr2000@yahoo.com, elharby@yahoo.co.uk, amriad2000@yahoo.com

Abstract: In this paper a new Natural Language Processing (NLP) system that has the ability to extract information from a database using rules and to search by the popular names. These names can be written as *fixed forms* or *variant forms*, for instance the name "ابراهيم" may be written correctly as "إبراهيم" or "ابراهيم" or wrong as "أبراهيم" for one person. The new search method is performed using Z-test, and we find that the new way has superior performance compared to the ordinary search.
[M.M. Badr, A.A. El-Harby and A.M. Riad. **New Searching Rules to Find Variant Names in Arabic.** J Am Sci. 2012;8(5):238-243]. (ISSN: 1545-1003). http://www.americanscience.org. 31

Keywords: Name Searching -NLP –understanding

1. Introduction

It is common that the words have more than one form of spelling in various languages such as in English. Many words can be written as a variant form but they have the same meaning. For example, "colour: color", "licence: license", "labour: labor", "metre: meter"… and so on . These words are saved in machine system as one form. So, when retrieving any of these words in another form, it is so difficult for the system to do that. Many search take This problem is Name Searching(NS) .

NS can be defined as the process of using a name as part of a query in order to retrieve information associated with that name in a database [1]. Another definition of name searching that ,It is useful to define for purposed of what is meant by name searching and related terminology and to describe the application areas for which name searching systems have been developed. In their comprehensive review article of personal name-matching applications categorize applications as being: 1) name authority control, 2) information retrieval, and 3) duplicate detection [2,3]

Borgman and Siegfried considered, the same person, but more generally the same institutional, geographical, or other proper-named entities as well. This determination might be made solely on the basis of a direct comparison of the two strings, or more knowledge might be used, e.g., models of a) variant spelling or representation of names,
b) keying errors, c) phonetic models, or d) record-linkage [4]. **Raghavan and Allan** proposed the use of approximate string matching techniques to normalize names in order to overcome the problem. They show how they could achieve an improvement if we could tag names with reasonable accuracy in ASR [5]. **Carter and others** [6] described and evaluated a simple and general solution to the handling of compound nouns in Swedish and other languages in which compounds can be formed by concatenation of single words. They solved using a principled grammar-based language-processing architecture; it is then possible to accommodate input in split-compound format. **Al-Onaizan and Knight** presented a novel algorithm for translating named entity phrases using easily obtainable monolingual and bilingual resources. They reported on the application and evaluation of this algorithm in translating Arabic named entities to English [7]. **Freeman and others** presented a solution to the problem of matching personal names in English to the same names represented in Arabic script. Standard string comparison measures perform poorly on this task due to varying transliteration conventions in both languages and the fact that Arabic script does not usually represent short vowels. Significant improvement is achieved by augmenting the classic Levenshtein edit-distance algorithm with character equivalency classes [8].

Maloney and Niv, described a fast high performance name recognizer for Arabic texts. It combines a pattern matching engine and supporting data with a morphological analysis component. The role of the morphological analysis in accurate name recognition is discussed [9]. Named entity recognition is nowadays an important task, which is responsible for the identification of proper names in text and their classification as different types of named entity such as people, locations, and organizations. **Shaalan and Raza** presented their attempt of the recognition and extraction of the most important proper name entity, that is, the person name, for the Arabic language. They developed the system, person name entity recognition for Arabic, using a rule-based approach. [10].

Modern standard Arabic is usually written

without diacritics. This makes it difficult for performing Arabic text processing. Diacritization helps clarify the meaning of words and disambiguate any vague spellings or pronunciations, as some Arabic words are spelled the same but differ in meaning. **Shaalan and others** addressed the issue of adding diacritics to undiacritized Arabic text using a hybrid approach. The approach requires an Arabic lexicon and large corpus of fully diacritized text for training purposes in order to detect diacritics. Case-Ending is treated as a separate post processing task using syntactic information. The hybrid approach relies on lexicon retrieval, bi-gram, and SVM-statistical prioritized techniques. [11]. **Stalls and Knight** are used name searching to translate names and technical terms from English into Arabic. [12]. **Abuleil** studied how extract names form Arabic text for question answering system. Therefore, the proper names in Arabic do not start with capital letter as in many other languages so special treatment is needed to find them in a text. He presented a technique to extract names from text by building a database and graphs to represent the words that might form a name and the relationships between them [13].

Now we take the problem in Arabic database name and define some rules using the structure of Arabic language. The problem is described in section 2, The proposed system is described in Section 3. In Section4, the implementation of this system is presented. The results and concoction are discussed in Sections 5,6. Section 7 show some fields can used this system.

2 Problem Description

The problem of this study is how to search about the Arabic words in any object (database "DB", document, file and so on). These words may be entered as a form inside the object or a variant spelling, as in some names in Arabic language.

It is known that when the user searches by ordinary search methods on the database by variant word (as spelling) the required information cannot be found by the machine. Therefore, it sometimes causes a lot of problems and a waste of time and efforts.

This problem occurs in any world language, there are words that may be written in many forms. When we save these words in any object (DB or other) only one form is stored for each one. In order to carry out this search using another form, the system fails to find it.

To solve this problem, we search and analyze the letters and words in Arabic language, and then we recognize the letters that cause the problem. To solve this problem a system is designed. This system contains new rules that depended on the previous study in Arabic language [15, 16]. The proposed system is an intelligent system. This system is a new search system which has the ability to understand the different forms of writing.

3 The Proposed System

The aim of the proposed system is to introduce a new searching method able to find a word regardless its form entered by the user. In general, searching technique is done by comparing two words alphabetically. The two words are considered the same if they have the same letters, otherwise they are different.

Although many words in the database are stored in certain form, the user gets unexpected result. This result is due to variant spelling. For instance, the word "**Ahmed**" can be written in two forms ("**أحمد**" or "**احمد**") when this word is stored in the database using certain form and the user type another form, the search process cannot retrieve the required result.

By studying the names and the letters in Arabic language [14, 15], we can determine the variant letters which have many forms. Arabic names are divided into five classes, which are presented in the next section. In order to build an intelligent system to understand the variant letters; this system should be designed according to rules. These rules help the system to recognize the required name.

The intelligent proposed system is capable to understand the variant or fixed letters in Arabic names. This system depends on some rules, and The some levels which define in NLP understanding [16].

3.1 Suggested Rules

The proposed system depends on defining some rules. These rules are deduced from studying the letters and words in Arabic language.

In Arabic language, the letter may have many shapes e.g. the "ا" can take the forms "أ- إ- آ- ا". The variant letters are " ه - ة– ؤ – و– ء – ئ – ى – ي – أ – إ– آ– ا- null" [14,15].

Also by studding the Arabic words, we find five kinds of variants

1- The word has no variant letters.
2- The variant letter is in the beginning of the word.
3- The variant letter is in the end of the word.
4- The variant letter is in the beginning and the end of the word.
5- The variant letter is within the word.

The proposed system combined these five kinds of variants. These rules are created as follow:

Rule 1: no variant such that
محمد , منصور , رضا , هيام ,.... وهكذا.

Rule 2: The variant letter is at the beginning such that
(أحمد – احمد) , (أشرف – اشرف) , (آلاء –الاء) , (ابراهيم -

إبراهيم) ،... وهكذا.

Rule 3: The variant letter is at the end such that
(هبة– هبه) , (لمياء –لميا) , (على– علي) , (سلوى– سلوا), (لوئ – لوى) ...و هكذا .

Rule 4: the variant letter is at both the beginning and the last such that
(اسماء – أسماء- اسما), ...و هكذا.

Rule 5: the variant letter is in between the word such that
(ميرفت – مرفت) , (جاكلين – جكلين) , (ميرهان – مرهان)...و هكذا.

3.2 The Levels Used

NLP contains many levels as mentioned above. The levels used in the proposed system are LEXICAL, SYNTATIC, SEMANTIC, PRAGMATIC, and EVALUATION.

The suggested system divides the name into parts "tokens" by using the parser. Any token is consisting of three parts SUFFICES, MIDDLE, and PREFACES where suffices and prefaces are Arabic letters. This process determines if the letter is fixed or variant. The grammar is responsible for this process.

The semantic is used to find out if the token belongs to the grammar rules or not. The system contains a function to determine if the letter is variant letter or not. In the proposed system, the lexicon is a database which contains the information about all names. Finally, after all levels are used, the evaluation determines if the name belongs to the database and displays its information or it gives a massage.

3.3 Illustrated Example

The levels and rules used in the proposed system are applied on real example as given bellow.
To search the name "احمد محمد عطية ابو رجب", the system does the next steps:

Step 1: The parser divides this name into "احمد", "محمد", "عطية", "أبو" and "رجب"

Step 2: The parser divides every token to prefix, mid, and suffix as shown in Table 3.1.

Step 3: The grammar determines if the letter is variant form or fixed form such that "ا" and "و" are variant letters. But " ب, ح,... " are fixed letters. The grammar is defined in Fig. 3.1. This type of the grammar is called context free grammar and it refers to CFG.

Step 4: the function called ***SEM*** is used to determine if the letter is variant or fixed and is given by (3.1) and it represents the semantic level as given in Table 3.1.

$$\text{SEM(letter)} = \begin{cases} 1 & \text{if the letter is variant} \\ 0 & \text{if the letter is fiexd} \end{cases} \qquad (3.1)$$

Final step: by using the previous levels and searching the different cases in the lexicon (database). We get the information about the name "احمد محمد عطية ابو رجب".

Table 3.1 The parser output also the semantic rules.

Prefix	semantic	Mid	Suffix	Semantic
Null	0	احمد	Null	0
ا	1	حمد	Null	0
Null	0	احم	د	0
ا	1	حم	د	0
Null	0	محمد	Null	0
م	0	حمد	Null	0
Null	0	محم	د	0
م	0	مح	د	0
Null	0	عطية	Null	0
ع	0	طية	Null	0
Null	0	عطي	ة	1
ع	0	طى	ة	1
Null	0	أبو	Null	0
أ	1	بو	Null	0
Null	0	أب	و	1
أ	1	ب	و	1
Null	0	رجب	Null	0
ر	0	جب	Null	0
Null	0	رج	ب	0
ر	0	ج	ب	0

$$\text{token} \rightarrow \begin{cases} \text{Null} + \text{Mid} + \text{Null} \\ \text{prefix} + \text{Mid} + \text{Null} \\ \text{Null} + \text{Mid} + \text{suffix} \\ \text{prefix} + \text{Mid} + \text{suffix} \end{cases}$$

$\text{prefix} \rightarrow \text{Variant}_1, \text{Variant}_2, \text{Variant}_3, \text{Variant}_4, \text{fixed}$

$\text{suffix} \rightarrow \text{Variant}_1, \text{Variant}_2, \text{Variant}_3, \text{Variant}_4, \text{fixed}$

$\text{Mid} \rightarrow \text{Variant}_1, \text{Variant}_2, \text{Variant}_3, \text{Variant}_4, \text{fixed}$

$\text{Variant}_1 \rightarrow$ ء,أ,إ,آ,ا

$\text{Variant}_2 \rightarrow$ و,ؤ

$\text{Variant}_3 \rightarrow$ ي,ى,ئ

$\text{Variant}_4 \rightarrow$ ة,ه,ت

$\text{fixed} \rightarrow$ any letter disapear privouse in Variant_1, Variant_2, Variant_3, and Variant_4 such that ع,ف,ب,.....

Fig. 3.1 The Grammar In Proposed System

4. Implementation

The system contains two parts as shown in Fig. 3.1.

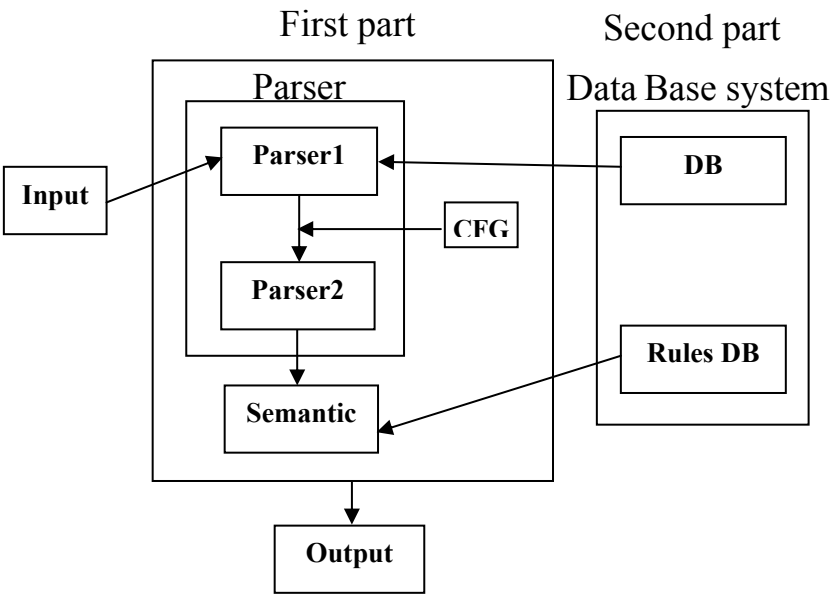

Fig. 3.1 The Propose System Component.

First Part is a database (DB) system; it contains the database about students in the faculty of science in Damietta and another about the rules in Arabic language (knowledge base).

Second Part is a project. This project contains some GUIs that work together with the database. One of these GUI contains the new searching technique as shown in Fig 3.2.

Fig 3.2. main GUI in the system

Button the Rules is connected to rule base and database also it contains the levels of NLP which used in the system. Button the Rules It allows the user to insert the searched name. The system divides the full name into the sub-names for example "احمد على الموجى " is divide into"احمد ", "على" and "الموجى " also the system is gain do the same process in every name in the database. Every part of the entered name and the part of the name in database is matched by on of the rules which known in the system. If at least one rule is exiting then, the system returns a number one, (means match success) else return zero (means match fail). By multiplying every returned value in all parts and checked it if one, it means this name is exiting and the result is shown but if the value is zero it means doesn't exist.

5 Results and evaluate the System

The proposed system has been designed and tested on 200 different names. Section 5.1 summarized the output of the proposed system and it give the results after applying the ordinary search and search by the rules, respectively. Section 5.2 is compared between the result of the two methods (search by rules method and ordinary search method).

5.1 Summarized of the Results

Table 3.2 contains sample of the names which has been searched in the database. This table shows that:-

1- In the case the required name is the same as in the database, the two search methods are equivalent and retrieve the name.
2- In the case the required name has variant letters at the beginning or at the end or both the search by rule method retrieves the name but the ordinary search method fails.
3- In the case the required name has variant letter in between the first and last letter the two search methods, search by rules and ordinary search, fail.

Table 3.2 Sample of names which search for and the results obtained by the two methods

Index	Required Name	Ordinary Search	Search by Rules
1	حسن على حسن البسيونى	حسن على حسن البسيونى	حسن على حسن البسيونى
2	سمر على حسن شعلان	سمر على حسن شعلان	سمر على حسن شعلان
3	**اسما** عمر مشعل زغلول	No result	**أسماء** عمر مشعل زغلول
4	**لميا** مجدى حلمى خميس	No result	**لمياء** مجدى حلمى خميس
5	**اسلام** فاروق عبد الهادى القلش	No result	**إسلام** فاروق عبد الهادى القلش
6	مها ممدوح عباس السيد	مها ممدوح عباس السيد	مها ممدوح عباس السيد
7	**رضوا** رضا **إبراهيم** سعيد	No result	**رضوى** رضا **ابراهيم** سعيد
8	**ملود** ضيف محمد **خمس**	No result	No result
9	**ميرفت** على جودة	No result	No result
10	**ففيان** عبد الباسط على حسن علام	No result	No result

5.2 Evaluate the system performance

To evaluate the system performance, we compare the results for ordinary search and searching by rules. Statistical hypothesis method (Z-test) is used. The system is tested using a random sample of 200 student's names. The following results are obtained.

Case 1: By using ordinary search the system knows 93 out of 200 names.

Case 2: By using search by rules the system knows 197 out of 200 names.

By using Z-test for the difference between the two proportions to compare the results obtained in the two cases. P_1=93/200=0.465, where P_1 means the proportion obtained from ordinary search. P_2 =197/200=0.985, where P_2 means the proportion obtained from searching by rules. we get the Z is 11.64576 but $Z_{0.05}$ is 1.645 , and $Z_{0.01}$ is 2.325.This result miens the new search more significant than ordinary search.

6. Conclusion

The proposed system is designed depending on new search rules. 200 random Arabic names. This names contains 60 variant names. The performance of this system is 97.5%. It was found that this system was better than the ordinary search methods.

The system, sometimes gives more than one result to the same name searching. Therefore the user should decide which result (data) will be dealt with. E.g; on searching the name "اميرة" the results will be whether "اميره" or "اميرة" or "أميرة". The result will be displayed if we search the first name. But, when the user searches for full name of a person (first name, father's name, and the title) the result is often the same. This result does not depend on the system but depends on the stored names in the database.

The performance rate for the test was 97.5%. The wasting data (2.5%) is explained by forgetting a letter in-between the first and the last letter in the name searching e.g. "شيرين" can be search by "شرين".

7. Application Fields of Search Method

It is known that, the computers are a basis tool in any institution. It is common using the databases and making search process. Therefore, the different proposed search methods can be applied inside any Arabic database to retrieve words in general. One can find database in many places such as passports departments, the airport databases, social affairs or students' databases, shopping databases, the institutions of health… etc. Also they can be added as a useful tool in the search engine into Web or it can be used in the search engine in personal computers. Moreover, they can be applied in translation process from Arabic into other languages, when the translation process depends on a dictionary and the user may insert a written name which is not found in the dictionary and in this case the system can not retrieve this name and the translation process fails. But by helping one of the suggested methods, the system can retrieve information and the translation process succeeds.

Corresponding author
A.M. Riad
Department of Information System, Faculty of Computers and Information Systems, Mansoura University, Egypt.
mohfbadr2000@yahoo.com
elharby@yahoo.co.uk,
amriad2000@yahoo.com

8. Reference

[1] P. Thompson and C. Dozier (1997), "Name Searching and Information Retrieval", Available in The Computation and Language (ACL).
[2] K. Kwok, and P. Deng (2002), "Corpus-Based Pinyin Name Resolution", Proceedings of the First SIGHAN Workshop on Chinese Language Processing (COLING 2002). pp. 41-47.
[3] T. DiLauro, G. Choudhury, M. Patton and J. Warner (2001), "Automated Name Authority Control and Enhanced Searching in the Levy

Collection", D-Lib Magazine Volume 7 Number 4.

[4] J. Gao, M. Li, A.Wu, and C. Huang(2006), "Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach", Association for Computational Linguistics, ACL.

[5] H. Raghavan and J. Allan (2004), "Using Soundex Codes for Indexing Names in ASR documents", In Proc. Human Language Technology conference - North American.

[6] D. Carter, J. Kaja , L. Neumeyer , M. Rayner, F. Weng, and M. Wiren (1996), "Handling Compound Nouns in a Swedish Speech-Understanding System", In ICSLP-, pages 26-29.

[7] Y. Al-Onaizan and K. Knight (2002), "*Translating Named Entities Using Monolingual and Bilingual Resources*" Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp. 400-408.

[8] A. Freeman, S. Condon and C. Ackerman (2006), "Cross Linguistic Name Matching in English and Arabic: A "One to Many Mapping" Extension of the Levenshtein Edit Distance Algorithm", Proceedings of the Human Language Technology Conference of the North American, pp. 471–478.

[9] J. Maloney and M. Niv(1998), "TAGARAB: A Fast, Accurate Arabic Name Recogniser Using High-Precision Morphological Analysis.", Proc. of the Workshop on Computational Approaches to Semitic Languages. Montreal, Canada.

[10] K. Shaalan and H. Raza (2007), "Person Name Entity Recognition for Arabic", Proceedings of the 5th Workshop on Important Unresolved Matters, pages 17–24.

[10] K. Shaalan, H. Abo-Bakr, and I. Ziedan (2009), "A Hybrid Approach for Building Arabic Diacritizer", Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages, pages 27–35.

[2] B. Stalls and K. Knight (1998), "Translating Names and Technical Terms in Arabic Text", Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages.

[12] S. Abuleil (2004) "Extracting Names From Arabic Text For Question-Answering Systems", In Proceedings of Coupling approaches, coupling media and coupling languages for information retrieval (RIAO 2004), Avignon, France. pp. 638-647.

[15] الشيخ عبد الغنى الدقر، (1984) "*معجم القواعد العربية*" . دار بيروت للطباعة و النشر،.

[16] د. أحمد طاهر حسنين و د. حسن شحاتة (1988), قواعد الإملاء العربي بين النظرية و التطبيق", مكتبة الدار العربية للكتب.

[5] J. Allen (1995), "Natural Language Understanding", the Benjamin/Cummings Publishing Company Inc.

4/29/2012