Enhance the Process of Tagging and Classifying Proper Names in Arabic Text

Saleem Abuleil and Khalid Alsamara

Faculty of MMIS, Chicago State University, 9501 S. king Drive, Chicago, IL 60628, USA
sabuleil@csu.edu, kalsamara@hotmail.com

ABSTRACT

Extracting and classifying proper names is a key to improving the efficiency and the performance of many applications in the area of natural language processing and text mining. Valuable information in the text is usually located around proper names. To collect this information we need to find the proper names first. By extracting proper names from the text we provide these applications with the proper names found in the text, their location, and some information about each. Proper names in Arabic do not start with capital letters as in many other languages so special treatment is needed to find them in a text. Little research has been conducted in this area; most efforts have been based on a number of heuristic rules used to find names in the text; some have used graphs to represent the words that might form a name and the relationships between them; and some have used statistical methods for this purpose. In this paper we present a new technique to extract names from text using a hybrid system based on both statistical methods and predefined rules. First we tag the proper name phrases in the text that may include names; second we use statistical methods to extract proper names from these candidate phrases; and third, we classify each proper name with respect to its major class and its subclass. We have developed a variety of rules and tested several different assumptions to accomplish the goals of this research

***Keywords**: Arabic Language, Proper Nouns, Tagging, Classification, Rules, Statistical Methods*

1. INTRODUCTION

Names play a very important role in many areas in natural language engineering, especially in question-answering systems, text summarization, text classification, information retrieval systems and information extraction according to Cowie and Lehnert, [7]. Rau [15] argues that names not only account for a large percentage of the unknown words in a text, but are also recognized as a crucial source of information in a text for extracting contents, identifying a topic in a text, or detecting relevant documents in information retrieval systems. As defined in the Message Understanding Conference [5], name recognition consists of identifying and categorizing entity names (person, organization, location), temporal expressions (dates and times), and some types of numerical expressions (percentages, monetary values and so on), which are considered to constitute up to 10% of written texts [6]. Among the different techniques used to process these data, we find some systems based on statistical methods, such as Hidden Markov Models [4], some based on strictly linguistic methods, which make use of grammar rules [12], and finally, some that combine rules and statistics [14].

Many researchers have attacked this problem in a variety of languages but only a few limited research projects have focused on natural language processing problems for the Arabic language. Mehdi [13] describes a computer system for syntactic parsing of Arabic sentences. The system is implemented using Definite Clause Grammar (DCG) formalism in Prolog. Ibrahim, Douglas, and Faahmy [8] have suggested a framework to deal with the morphology of the Arabic language. Foxley and Feddag [10] adopted a strategy of combining affixes to alleviate the operation overhead of affix manipulation routines. Feddag and Foxley [9] provided a single powerful framework for an intelligent database where the system stores only the roots of the verbs and uses a program intelligent enough to handle all derived forms automatically.

Wacholder et al. [16] analyzed the types of ambiguity - structural and semantic - that make the discovery of proper names in the text difficult. Kim and Evens [11] built a natural language processing system for extracting personal names and other proper nouns from the *Wall Street Journal*. Yangerber et al. [17] presented an algorithm, called NOMEN for learning generalized names in text. NOMEN uses a novel form of bootstrapping to grow sets of textual instances and of their contextual patterns. Abuleil and Evens [2] built a parser that use a set of rules to parse the Arabic text, tag the proper nouns, and extract information about them. Abuleil [1] uses the relationships between the words in the proper name phrases by building a directed graph that represents the words as nodes and the relationships between them as weights on the edges.

2. PROPER NAMES IN THE ARABIC LANGUAGE

The problem of identifying proper names is particularly difficult for Arabic, since names in the Arabic language do not start with capital letters so we cannot tag them in the text by looking at the first letter of the word. To tag proper names in Arabic text we use keywords to guide us to the place where we can find them in the text. By using keywords we tag proper name phrases that might contain a certain name then we process these phrases to tag names. We discovered from our analysis of the

Arabic text that proper names can be classified into to different categories: people name, location name, organization name, product name, disease name, activity name, etc., with respect to the way they appear next to the keyword

3. RULES FOR TAGGING PROPER NAME PHRASES IN THE ARABIC TEXT

We generated a set of rules to predict where the names are located in the text. These rules are based on two things: special nouns and special verbs. We will refer to the special nouns as n-keywords and to the special verbs as v-keywords in this paper. Well-known names seem to appear close to one of these noun keywords or verb keywords in Arabic text. We collected tens of keywords in a previous research project [2] and we classified them into different classes: people, locations, organizations, diseases and products. Tables 1 and 2 show some examples of these keywords.

Table 1 V-Keywords

Keyword	Main Type	Sub Type
تحدث Said	Person	N/A
صرح Announced	Person	N/A

Table 2 N-Keywords

Keyword	Main Type	Sub Type
مراسل Reporter	Person	Reporter
رئيس President	Person	President
شارع Street	Location	Street
مدينة City	Location	City
جامعة University	Organization	University
شركة Company	Organization	Company
سيارة Auto	Product	Auto
مرض disease	Disease	N/A
موتمر Conference	Activity	Conference
في / ب in / at	Location	N/A
غرب West of	Location	N/A

Following are some rules we generated for this purpose:

Rule#1: n-keywords are used to tag personal names, organization names and location names while v-keywords used just to tag personal names.

Rule #2: Personal names may come either to the left or to the right of an n-keyword. If they appear to the right, they are attached directly to the n-keyword but if they appear to the left the name and the keyword can be separated by at most two words. In most cases the longest name is three words so we examine up to five words to the left of the n-keyword and three words to the right of the n-keyword to identify the proper name phrase. (In reading these diagrams please remember that Arabic is written from right to left.)

w5 …w2 w1 [n-k/w (people)] w3 w2 w1

Rule #3: When a personal name is attached to a v-keyword, it comes directly next to it and in most cases the longest name is three words so we examine three words to the right and three words to the left of the v-keyword.

w3 w2 w1[v-k/w (people)] w3 w2 w1

Rule#4: Organization and activity names come directly to the left (right after) the n-keyword. The longest name is five words so we examine five words to the left of the n-keyword to identify the proper name phrase.

w5 w4 w3 w2 w1 [n-k/w (org | activity)]

Rule#5: Location names come directly to the left of the n-keyword. The longest name is three words so we examine three words to the left of the n-keyword to identify the proper name phrases.

w3 w2 w1 [n-k/w (location)]

Rule#6: Product and disease names come directly to the left after the n-keyword. The longest name is two words so we examine two words to the left of the n-keyword to identify the proper name phrases.

w2 w1 [n-k/w (product | disease)]

Rule#7: More than one keyword can be mentioned in the same proper noun phrase, often a person-keyword followed by an organization-keyword. In this case we examine three words to the right of the keyword and eight to the left of the keyword.

w8 …w2 w1 [n-k/w(org) n-k/w (people)] w3 w2 w1

Rule#8: A proper noun phrase terminates when it encounters a stop word: a particle, verb, adverb, punctuation tag, etc., excluding the ones that are used as keywords

Rule#9: n-keywords that tag person names fall into two types: they either start with the letters "ال" meaning "the" in English (title keyword) or they do not (occupation keyword). If they start with "ال" most of the time the names come after the keyword immediately, but if they do not, most of the time an organization name appears between the keyword and the person

name. Examples: المدير شاكر Manager Shaker, مدير مركز القدس شاكر Manager of Al-Quds Center Shaker.

Rule#10: Organization, location, product and disease keywords, when they start with the letters "ال" meaning "the" in English, do not follow with a proper name but instead they follow with an adjective or an adjective derived from a proper name, such as الدولة الفلسطينية Palestinian State, with a few exceptions such as المملكة العربية السعودية Kingdom of Saudi Arabia.

Rule#11: Some keywords consist of two words. For example, the word "نائب" "Vice" is usually connected to the word "الرئيس" "President" to form the keyword " نائب الرئيس" "Vice President."

Rule#12: When an organization name consists of more than one word and when the second word in the proper name starts with letters "لل" meaning "for" in English, the first word is classified as a primary proper name and the rest of the words in the organization name are classified as co-proper names. Example: مصنع القدس *للفرشات* Jerusalem Factory for *Mattresses*.

Rule#13: When a noun appears between keyword and proper name we classify it as a co-keyword. Example: وزير *الزراعة* علي Minister of *Agriculture* Ali

Rule#14: Each word in a person name represents an independent name while each word in other proper name types should be mentioned along with other words around it to classify the whole string as a proper name. Examples: الدكتور سليم ابوليل Dr. Saleem Abuleil, موسسة الارض المقدسة Holy Land Foundation. Saleem and Abuleil each one of them represents a proper name even if they are not mentioned together, while the words "Holy" and the word "Land" must appear together before we can classify them as one proper name for the "Foundation"

4. METHODS OF TAGGING PROPER NAMES

After we identify and examine the proper noun phrases in the text, the next step is to tag and extract the proper names. A variety of different methods have been implemented and used for this purpose: Rule-based methods, graph-based methods, and statistics-based methods.

Rule-based methods [2] use a bunch of heuristic rules to parse text and tag the proper names. This technique has many limitations: it is hard to tell exactly where the name starts in the phrase and where it ends. We cannot tell, even if there is a proper name, whether it is attached to the keyword or not and if it is to the left of the keyword or to the right. No matter how many rules you add to the system you will never cover all the scenarios that you might face, since each person writes in a different way with a different style, so the same name phrase can be written in many different ways.

Graph-based methods [1] use the relationships between the words in the proper name phrases to build a directed graph that represents the words as nodes and the relationships between them as weights on the edges. The relationship (weight) between two words represents the number of times these two words appear attached to each other in the name phrases. This approach proved to give better results than our rule-based method, especially for organization and location names, but after we process a few hundred proper noun phrases the graph becomes complicated, and the more proper noun phrases there are to process, the more complicated the graph becomes and the harder it is to maintain and manage.

5. OUR APPROACH TO TAGGING PROPER NAMES

In this paper we use a hybrid system to tag and extract proper names by combining three different techniques: rules, graphs, and statistics. We use rules to tag proper noun phrases, we use a variant of the graph-based technique to locate full or partial candidate proper names by breaking proper noun phrases into tokens, where each one is either an individual word or two adjacent words, and we use some rules and the token frequency to identify proper names. For this purpose we use two main files, one to save tokens and one to save proper names as follows:

Tokens File

Token	PNP#	Status	Freq

Proper Nouns (PNs) File

PN

Token: either individual word or two adjacent words mentioned in a proper name phrase "PNP"
PNP-Code: A sequence number is generated and assigned to each new PNP.
Status: "Y" means a proper name or part of a proper name. "N" means that this token is not a proper name or part a of proper name.
Freq: number of times the token has appeared since the first time it was captured.

This method carries out the work in three steps: prepare a list of candidate proper noun phrases, update the tokens file, and clean up the tokens file. First, when we receive a new PNP we assign it a unique code and break it down into tokens based on the keyword(s) mentioned in it as follows:

Keyword type: *Person*
Tokens: W1, W2, W3, ..., Wn

Keyword type: *Organization, location, activity, product, and disease*
Tokens: W1,
W1 + W2, W2 + W3, …, Wn-1 + Wn

Then we check each token to see if it was previously tagged as a proper name or not:

For each token (Ti) do:
If Ti is a PN → tag Ti "PN"

where i: 1..n, n: number of tokens found in the proper noun phrase

Second, we update the token file by checking each new token against all tokens in the token file. If there is a match we increment the token frequency by one. If not we add it to the file as a new entry. If there is a match and if the result of dividing the frequency of the token by the number of words in the token is greater than a threshold value "n1" we change the status of the token to "Y."

For each Ti do
For each entry "token" (Tj) in the token file do:
If (Ti = Tj) →
Increment Tj_Freq by one
Tag Ti "Found"

If (Ti = Tj) and (Tj_freq / | Tj | > n1)
and (Tj_status = "N") →
Turn Tj_status to "Y"

If Ti is not tagged "Found" and Ti is not tagged "PN" →
Create a new entry for Ti:
(Ti, pnp-code, N, 1)

If Ti is not tagged "Found" and Ti is tagged "PN" →
Create a new entry for Ti:
(Ti, pnp-code, Y, 1)

Third, frequently we clean up the tokens file and update the proper names file by performing several tests:

1- For each proper noun phrase in the token file, we classify its tokens into two classes: proper names and non-proper names. If the frequency of the non-proper name is less than frequency of the proper name by a threshold value "n2"we drop the non-proper noun token from the file:

For each PNPi in the token File
For each Token belongs to PNPi do:
If freq (Token) / freq (Tk) <= n2 →
Drop the entry "Token" from the token file

Tk: Min [Freq [All Tokens belong to PNPi with status = "Y"]]

2- If all tokens belong to one particular proper noun phrase are classified as proper names we use the following rules to identify the final version of the proper names, save them it in the proper name file and drop them all from tokens file:

Person names:

- If W1 is a person name (first name), W2 is a person name (last name) then Freq (W2) >= Freq (W1).

- If Wn-1 is a person name and Wn+1 is a person name then Wn is a person name.

- If a person name consists of two words, the first word is considered to be a first name, the second word is considered to be a last name and if a person name consists of one word it is considered to be a last name.

Other proper name types:

- If W1 is tagged as a proper name and W1 + W2 is tagged as proper name then ignore W1 and consider W1 + W2 as a proper name. Example:
PNP: جامعة القدس المفتوحة Al-Quds Open University
Tokens: القدس المفتوحة / القدس Alquds / Alquds Open.

- If Wn-1 + Wn tagged as proper name and Wn + Wn+1 tagged as proper name then ignore Wn-1 + Wn and Wn + Wn+1 and consider Wn-1 + Wn + Wn+1 as proper name. Example:
PNP: الأمارات العربية المتحدة United Arab Emirates
Tokens: الأمارات العربية / العربية المتحدة United Arab / Arab Emirates

3-If none of the tokens that belong to the same proper noun phrase are classified as proper nouns after the "r1" period then we drop them all along with the proper noun phrase from the tokens file. "r1" is a threshold value represents the difference between the code of the proper noun phrase we are checking and the code of the last proper noun phrase captured.

6. PROPER NAME CLASSIFIER

Some names may be attached to different types of keywords and to more than one keyword in the same name phrase. Examples:

السيد عباس **رئيس السلطة** الفلسطينية
Mr. Abbas the **President** of Palestinian **Authority**

رئيس جامعة القدس المفتوحة الدكتور يونس عمرو
President of Al-Quds Open **University Dr.** Younis Amro

After we find the name we classify it with respect to its major class and its subclass:

Major class: person, organization, location, product, etc.

Sub-class: president, mister, commander, professor, bank, store, city, state, camp, etc.

We use the following equations to classify the names:

$$pos(Name \mid KWi) >= R2$$

and

$$\frac{pos(Name \mid KWi)}{pos(Name \mid KWi) + neg(Name \mid KWi)} >= R3$$

Where:

pos (Name | KWi): the number of times the name found is attached to the keyword KWi.

neg (Name | KWi): the number of times the name is found attached to keywords other than KWi.

7. EXPERIMENTAL RESULTS

We have tested our system on 300 articles from the *Al-Quds* newspaper [3], published in Palestine. The system extracted 4603 proper noun phrases classified into 1795 person name phrases, 1601 organization name phrases, 938 location name phrases, 66 product name phrases, 193 activity name phrases and 10 disease name phrases. We tested both the Name Tagger method and the Name Classifier method. For the first method we used the following threshold values n1, n2, r1 respectively 2, 0.5, and 1000. The module identified 3081 names (297 distinct names), missed 50 names, and extracted 37 names mistakenly out of 1522 garbage proper noun phrases (keywords with no proper names around them). We found that most of the proper name phrases tagged to the right of a person keyword are garbage proper name phrases. Table 3 shows the number of extracted names (tokens), distinct extracted names (types), and missing names for all proper name types. Table 4 shows the number and the percentage of names extracted and the number and the percentage of names missed by the Name Tagger Method.

The reason for the missing names is that these strings were not mentioned enough times to qualify them as names. When we checked the token file we found all the missing names there but their weight (frequency) was insufficient to qualify them as names. The system could not extract the names mentioned in the document with no keywords attached to them. Fig. 1 shows how the performance improves as the system processes more articles and stores more information in the tokens table. The proper noun phrases are grouped into ten groups, 460 proper noun phrases in each one to show the total number of names extracted and not extracted in each group of proper noun phrases

In Fig. 2 the proper noun phrases are grouped into ten groups, 460 proper noun phrases in each one, to show the comparison between the three methods for extracting the proper names in the text: the new technique "Hybrid System" we use in this paper, the system built by Abuleil and Evens [2] based on heuristic rules, and the system built by Abuleil [1] based on graphs to represent the relationships between words in the proper noun phrases. The figure shows the total number of names extracted by each method in each group.

Table 3 Comparison between Different Types of Proper Names

PN Type	# of Names Extracted	# of Distinct Names Extracted	# of Names Missed
Person	1287	102	11
Location	450	54	14
Organization	1279	123	18
Activity	48	13	4
Product	17	5	1
Disease	0	0	2
Total	3081	297	50

Table 4 Comparisons between Proper Names Captured and Not Captured

PN Type	# & % Distinct Names Captured	# & % Names Missed	Total
Person	102 90.3%	11 9.7%	113
Location	54 79.4%	14 20.6%	68
Organization	123 87.2%	18 12.8%	141
Activity	13 76.5%	4 23.5%	17
Product	5 83.3%	1 16.7%	6
Disease	0 0%	2 100%	2
Total	297 85.6%	50 14.4%	347

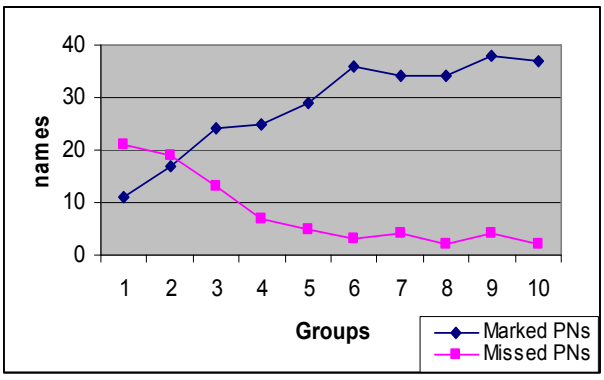

Fig.1 Tagged names vs. Untagged Names

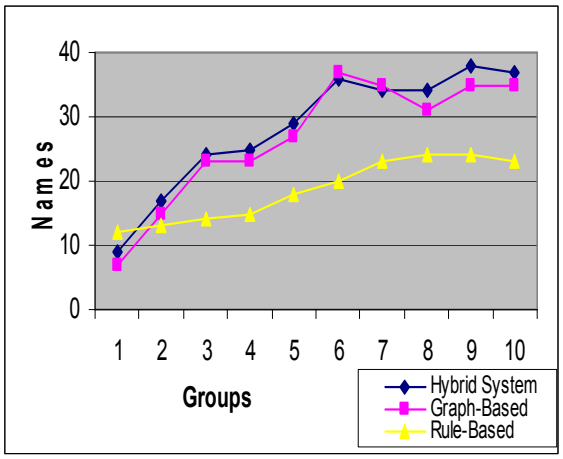

Fig.2 Comparison between Different Methods

We classified the names according to two types: major class (people, location, organization, and event) and specific subclass (president, country, newspaper, war, etc.). We used the following values for r2 and r3 respectively, 3 and 0.7. Table 5 shows the number of names classified correctly and the number of names not classified correctly. One major reason for the misclassification of names is that some person names appear in a phrase that contains both a title and an organization. The system achieved 100% accuracy when it classified names with respect to the subclasses.

Table 5: Classification with respect to Major Classes

Major Class	# & % Names Captured	# & % Classified correctly	# & % Not Classified correctly
Person	1287	1268 98.5%	19 1.5%
Organization	1279	1264 98.8%	15 1.2%
Location	450	450 100%	0 0%
Activity	48	47 97.9%	1 2.1%
Product	17	100%	0%
Disease	0	0%	0%
Total	3081	3028 98.3%	36 1.7%

Fig. 3 shows the time required to process the proper noun phrases using both the hybrid system and the graph-based system. The graph-based system becomes complicated and needs more time when the graph becomes large.

8. CONCLUSION

We have described a new system to extract names from Arabic text by collecting information about the words in the text. We built a hybrid system using three techniques: rules, statistics, and relationships between words. We have tested our new system on 4603 proper noun phrases. We extracted 98.4% of all names and 85.6% of the distinct names found in the text. We found all of the missing names in the token file where we collect the words in the proper noun phrases so we believe that if we run more data where these names appear the system will extract them.

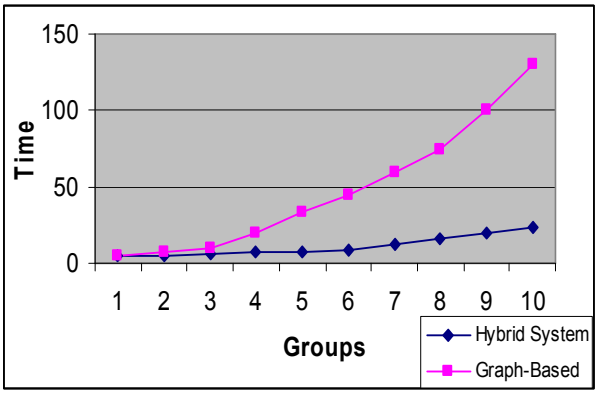

Fig. 3: Comparison between Different Methods with respect to the Time Required

REFERENCES

[1] Abuleil, Saleem, 2004. "Extracting Names From Arabic Text for Question-Answering Systems". *RIAO'04, Proceeding of the 7th International Conference on Coupling Approaches, Coupling Media, and Coupling Languages For Information Retrieval.* University of Avignon (Vaucluse), France April 26th-28th, 2004. pp 638-647.

[2] Abuleil, S. and Evens, M., 2002. Extracting an Arabic Lexicon from Arabic Newspaper Text. *Computers and the Humanities*, 36(2), pp. 191-221.

[3] *Al-Quds Newspaper,* 2005. Palestine.

[4] Bikel, D., Miller, S., Schwartz, R, Weischedel, R. 1999, An algorithm that learns what's in a name. *Machine Learning: special issue on Natural Language Learning,* 34: 211-231.

[5] Chinchor, N., 1998, Overview of MUC-7. In *Proceedings of the 7th Message Understanding Conference (MUC-7).*

[6] Coates-Stephens, S., 1992, The analysis and acquisition of proper names for robust text understanding. Ph.D thesis, Department of Computer Science, City University, London.

[7] Cowie, J., and Lehnert, W., 1996, "Information Extraction", *Comm of the ACM*, 39(1) 83-92.

[8] Ibrahim, A., Douglas, J., and Faahmy, A., 1989. "Arabic Machine Translation". *Proceedings of the First International Conference on Computing in Arabic-English,* Cambridge University, UK.

[9] Feddag, A., and Foxley, E., 1991. "An Intelligent Lexical Analyzer for Arabic and English". *13th Research Conference on Information Retrieval.* British Computer Society *(BCS).* Lancaster University, 8-9th April, UK.

[10] Foxley, E., and Feddag, A., 1990. "A Syntactic and Morphological Analyzer of Arabic Words". *Proceedings of the Second International Conference on Computing in Arabic-English*. Cambridge University, UK.

[11] Kim, J-S., and Evens, M., 1995, "Extracting Personal Names from the Wall Street Journal", *Proceedings of the 6th Midwest Artificial Intelligence and Cognitive Science Society Conference*, Carbondale, IL, April 21-23, pp. 78-82.

[12] Magnini, B., Negri, M., Prevete, R., Tanev, H A., 2002, WordNet Approach to Named Entity Recognition. *Proceedings of the Workshop, SemaNet'2002.* Binding using semantic networks.

[13] Mehdi, S. A. 1986. "Arabic Language Parser", *International Journal of Man-Machine Studies*. 2(5):593-611.

[14] Mikheev, A., Grover, C., Moens, M. 1998 Description of the LTG system used for UMC-7 *Proceedings of Message Understanding Conference (UMC-7).*

[15] Rau, L. F., 1991, "Extracting Company Names from Text", *Proceedings of the Seventh Conference on Artificial Intelligence Applications*, Feb. 24-28, Miami Beach, Florida, pp.29-32.

[16] Wacholder, N., Ravin, Y., and Choi, M., 1997, "Disambiguation of Proper Names in Text", *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Mar 31- Apr 3, Washington, DC, pp. 202-208.

[17] Yangerber, R., Winston, L, and Grishman, R., 2002, "Unsupervised Learning of Generalized Names", *COLING 2002*, Taipei. pp.1135-1141.