DEVELOPMENT OF ARABIC NOUN PHRASE EXTRACTOR (ANPE)

Islah K. Gharaibeh[1] and Natheer K. Gharaibeh[2]

[1] Prince Abdullah Bin Ghazi Faculty of IT, Al-Balqa Applied University, Salt, Jordan
[2] College Computer Science and Engineering, Taibah University, Yanbu, Saudi Arabia

ABSTRACT

Extracting key phrases from documents is a common task in many applications. In general: The Noun Phrase Extractor consists of three modules: tokenization; part-of-speech tagging; noun phrase identification. These will be used as three main steps in building the new system ANPE, This paper aims at picking Arabic Noun Phrases from a corpus of documents, Relevant criteria (Recall and Precision), will be used as evaluation measure. On the one hand, when using NPs rather than using single terms, the system yields more relevant documents from the retrieved ones, on the other hand, it gave low precision because number of the retrieved documents will be decreased. At the researchers conclude and recommend improvements for more effective and efficient research in the future.

KEYWORDS

Information Retrieval (IR), Natural Language Processing (NLP), corpus , Semitic languages , Arabic Noun Phrase Extractor (ANPE) , Noun Phrase (NP), Tokenization, Tagging, Parsing, Recall, Precision , inverted file.

1. INTRODUCTION

Arabic is considered the most widespread Semitic languages in terms of native speakers, it is the mother tongue of over 300 million people around the world [1], The number of Arab-speaking Internet users in 2016 was more than 155 million, about 43.4 % of the Arab world population [2]. But on the other side of the picture it is known as one of the most difficult languages in Natural Language processing (NLP) in general and information retrieval [3]. Further , there was an old debate and big disagreement between the oldest two Arabic grammar schools (Basra and Kufa) in one of the most important issues in Arabic language : the nominal sentence or noun phrase [4] , in spite of these difficulties and debate the researchers decided to overcome this issue in previous paper [5] . which presented a plan for building ANPE, Whereas This paper aims to implement ANPE as described by its predecessor [5].

However, the previous paper provided Introduction to the concept of Noun Phrase (NP) in general and the reason for Extracting NPs, it shews Literature Review, provided a Structure of Arabic Language by presenting these points:

1. Historical Survey of Arabic Grammar
2. Challenges of Arabic in NLP and IR
3. Arabic Tagset
4. Nouns in Arabic Language
5. Sentences in Arabic Language

DOI: 10.5121/ijnlc.2017.6105

The list of topics that is presented in this paper is to complete the previous one, it specifies more on the methodology used for building ANPE: tokenization; part-of-speech tagging; and noun phrase identification. further, it shows the resulted system of ANPE and its Run, and finally it explains the Evaluation and discuss the results.

In the next subsection, we will give a brief Background of the topic. Section 2 shows the Methodology, Section 3 provide the Implementation of ANPE, Section 4 presents the Running the ANPE System, then the Evaluation of IR System will be given in Section 5, and finally the Conclusion.

1.1 Background

A noun in Arabic is a word that indicates a meaning by itself without relating to the notion of time [6]. There are two main kinds of noun: variable and invariable. The grammar specifies the structure of the Arabic sentence. The Arabic sentence is generally classified as either nominal sentence or verbal sentence [7]. Accordingly, a nominal sentence is defined as "a sentence which begins with a noun, whereas a verbal sentence is one which begins with a verb". in this paper, we want to manipulate nominal sentences or Noun Phrases (NPs). NPs are good indicators of text content [8], “It has always been assumed by researchers that in language it is the noun phrases that are the content-bearing units of information”.

However, as presented in the next section, manipulating and identifying NPs is not trivial job, Once the text has been tokenized and divided into sentences, it is ready for the extraction process. The essence of this process is part-of-speech, discovering higher level structures and patterns in the text, associating each word with its grammatical context.

In searching about manipulating Noun Phrases, we found that at the beginnings, the used approaches targeted simple Finites State Machines and other compiler techniques, e.g., LR parsing. But after the revolution of Web2.0 and the increased interest in corpus and big Data there become a need for more intelligent ways for analysing data, Since then, many effective Machine Learning approaches [9] have been presented:

- Transformation-based Learning
- Memory-based Learning
- Maximum Entropy Model
- Hidden Markov Model
- Support Vector Machines

The goal of this paper was to develop a system for Extracting Arabic Noun Phrases, the next section will show more details about the system.

2. METHODOLOGY

This section discusses the steps of developing the proposed system: the Arabic Noun Phrase Extractor (ANPE) [5], including exhibition for the steps of the design and implementation of the system.

As mentioned in the introduction that most of the Noun Phrase extractors use three main steps:

1. Tokenization
2. Tagging
3. Forming Noun Phrases

Our system also followed these steps in its design – as seen in Figure.1–, so that they will be explained in details in the following subsections.

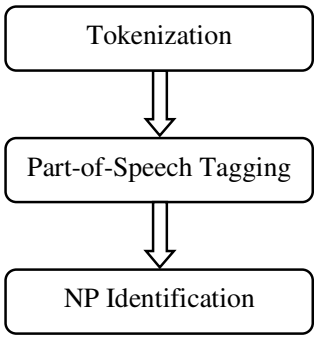


Figure.1: General steps of the ANPE system

2.1 Tokenization

This step is one of the most critical phases used to analyse text linguistically. "It breaks strings of characters, words, spaces and punctuation into tokens during the indexing process" [10]. The goal of the tokenization process is defined in [11] as "to determine sentence boundaries, and to separate the text into a stream of individual tokens (words) by removing extraneous punctuation. Document texts must be tokenized correctly, to enable the noun phrase extractor to parse the text efficiently". Also, as [12] illustrates that "Tokenization is a process to separate the texts into individual symbols. It adds that this is a particularly important issue for the types of texts that are used for concept space generation. Each text document must be correctly tokenized before the noun phrases are extracted. Otherwise, the noun phrase will be incorrect".

But first, it is necessary to Filter Out Stop Words from the documents so that the rest words in them are either nouns or verbs.

2.1.1 Stop Words Filtering (or stripping)

In English, stop words include articles such as "the," "a," and "an," and demonstratives like "this," "that" and "those." Removing these commonly occurring words from indices reduces the number of words each search term must be compared against, significantly improving query response time without affecting accuracy. These vary with each locale, because every language has different stop words [10].

Likewise, in Arabic Stop Words includes any word that is not considered part of speech, i.e. noun or verb (including prepositions (...الى، عن، في،), demonstratives (...هذا، هذه، هذان،), special characters ($, %, &,…), adverbs (...فوق، تحت،), … etc.)

2.2 Part-of-Speech Tagging

Resuming what had been discussed in [5] about Arabic tagset, here we built a tagger that is a system that outputs the part of speech of a given word, precisely whether it is a noun or a verb, so that we can exclude the verbs and extract only the nouns from the text.

The following steps illustrates how our tagger works:

- Search the word which we want to classify in the manually created D.B., if it was present so directly retrieve its classification, but if not move to next step.

- Examining the word:
  If begins with ال , or
  The first letter was م and last one ة , or
  The last letter was ة , or
  The last letter was اء
  Then the word is considered to be a noun.

- Examining the preceding word (words that is followed certainly with a noun), then searching this word from any groups that precede the noun.

- if any of the previous rules didn't match the word so we count the word's letters to know pattern for this word, then examining this pattern if it belongs to any of the noun patterns (stored in a table).

2.3 Noun Phrase Identification

In English Noun phrases are extracted using a finite set of rules, composed of different sequences of part-of-speech tag patterns such as the grammar shown below in Figure.2.

NP → ART NP2
NP → NP2
NP2 → N
NP2 → ADJ NP2
NP2 → NP3 PREPS
NP3 → N
PREPS → PP
PREPS → PP PREPS
PP → NP

Figure.2 : An example of a NP identification grammar in English [13]

The Noun Phrase formation system used in this study (which concerns about Arabic Language), builds two-term phrases according to the rule : NP → N N , which means that the Noun Phrase is constructed from two noun terms identified by the tagger, such that the tagger outputs a table of the nouns (as shown in Figure.10.b), then the Noun Phrase is identified by taking every two adjacent nouns together (as shown in Figure.3.c).

When looking to the example in Figure.3, for the sample sentence of (a), such a strategy produces the phrases in (c), We note that the phrase generation system identifies apparently reasonable constructions such as "تكنولوجيا المعلومات", "الحاسب الإلكتروني" and "جامعة اليرموك", but not the unwanted phrases "اليرموك علوم"

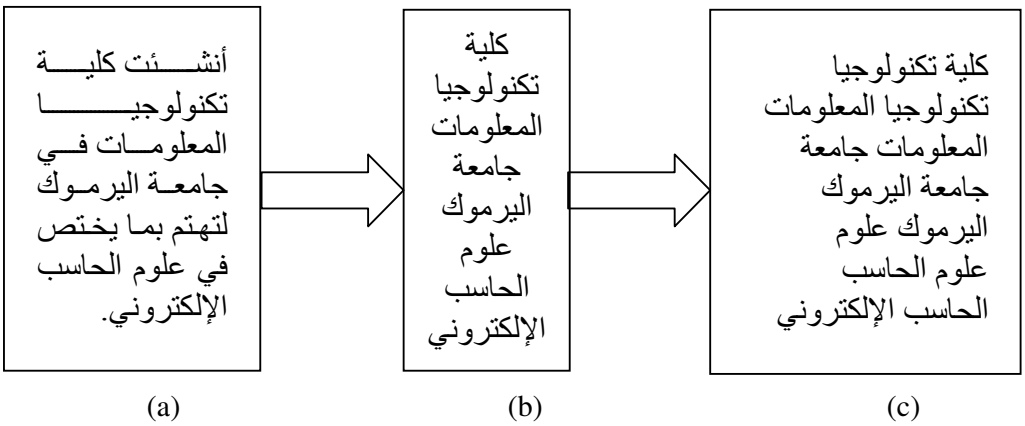


Figure.3: A simplified example of identifying NPs in the system

2.3.1 Automatic Phrase Assignment

An automatic phrase construction system generates a large number of phrases for a given text [14]. "The output could be used directly in a semi-automatic indexing environment by letting the user choose appropriate index entries from the available list. The standard entries from the output might then be manually chosen for indexing" [14].

In a fully automatic indexing system, Salton in [14] said that "additional criteria must be used, leading to the choice of some of the proposed phrase constructions, and the rejection of some others".

But the criteria he mentioned isn't agreeable with our system for many reasons, such as :

- His phrase formation system is different from ours, such that his system builds two-term phrases by taking all the possible combination of the terms.
- He handled also the phrases consisting of more than two words.
- The different nature of Arabic from English, such as nouns in Arabic could be adjectives or adverbs, as shown in the fourth section from section 3.

So, that we need a further choice of phrases, as well as a phrase ordering system in decreasing order of apparent desirability, and this can be achieved by assigning a phrase weight to each phrase and listing the phrases in decreasing weight order (ranking the phrases). In order to do so we need two different frequency criteria:

- The frequency of the term in a given document, known as the term frequency (tf)
- The number of documents, in which the term occurs, known as the document frequency (df).

The corresponding term weighting system, known as tf-idf is computed by the following formula (which give us the weight of a specific term i within a specific document j):

$$W_{i,j} = \frac{tf_{i,j}}{\max(tf_{i,j})} \times log\frac{N}{df_i}$$

(This will be explained briefly in building the inverted file mentioned later in section 3)

To get the term weight (W_i): we take the average weights of the term i (summation of the term weights within all the documents that it appears in divided by number of documents that it appears in)

Then we calculate the weight of the phrase (constructing from term i and term j) by taking the average weight of the two terms (i, j) that constructed it, as:

$$Phrase\ weight = \frac{W_i + W_j}{2}$$

Here we have a reasonable indexing policy consists in choosing phrases, after we ranked phrase list in decreasing order per a phrase weight. Using such an ordered list, a typical indexing policy consists in choosing the top n entries from the list, or choosing entries whose weight exceeds a given threshold T –As seen in Figure 11 in section 4.

3. IMPLEMENTATION OF ANPE

By the great expansion of the World Wide Web (WWW) and the presence of inexpensive user interfaces and mass storage devices in the recent years, Information Retrieval (IR) has changed considerably, such that IR systems become in need to an effective method for dealing with the vast amount of information through following an efficient way of indexing for the vast amount of information.

Here we proposed a phrase extracting system. It is desktop application which is built using Visual Basic 6 (VB6). In this section, we described the effect of using such technique in the Information Retrieval System and introduced the results and evaluation of the system.

3.1 Building the IR System

The following steps have been used to implement the IR System:

Step.1: Remove all of the stop words from the documents.
Step.2: Build the inverted file for the documents.
Step.3: Choose a query (target query) from the source query list.
Step 4: Begin the search for the relevant documents to the selected query, the search will be done on the inverted file (for more details read the part talking about the inverted file).
Step 5: Use the cosine similarity formula (given below) to determine the similarity between the Query and the retrieved documents:

$$Sim(D_i, Q) = \frac{\sum_{k=1}^{t}(d_{ik} \times q_k)}{\sqrt{\sum_{k=1}^{t} d_{ik}^{\ 2} \times \sum_{k=1}^{t} q_k^{\ 2}}}$$

The searching process (in step.4) is done in a special data structure – the inverted file (built in step.2), which has the following information:

- The frequency of each index term in the corresponding document (tf).
- Number of documents in which the index term appears (df).
- The maximum frequency that is computed overall index terms mentioned in the document.
- The document number.
- The index terms in each document.

- The weight of each index term in the document, calculated according to the following formula:

$$W_{ij} = \frac{tf_{ij}}{\max(tf_{ij})} \times log\frac{N}{df_i}$$

And the weight of the query is calculated according to the following formula:

$$W_{iq} = 0.5 + 0.5 \times \frac{tf_{ij}}{\max(tf_{ij})} \times log\frac{N}{df_i}$$

4. RUNNING THE IR SYSTEM

By running the system will start by showing the general interface as in figure 4 :

ذ

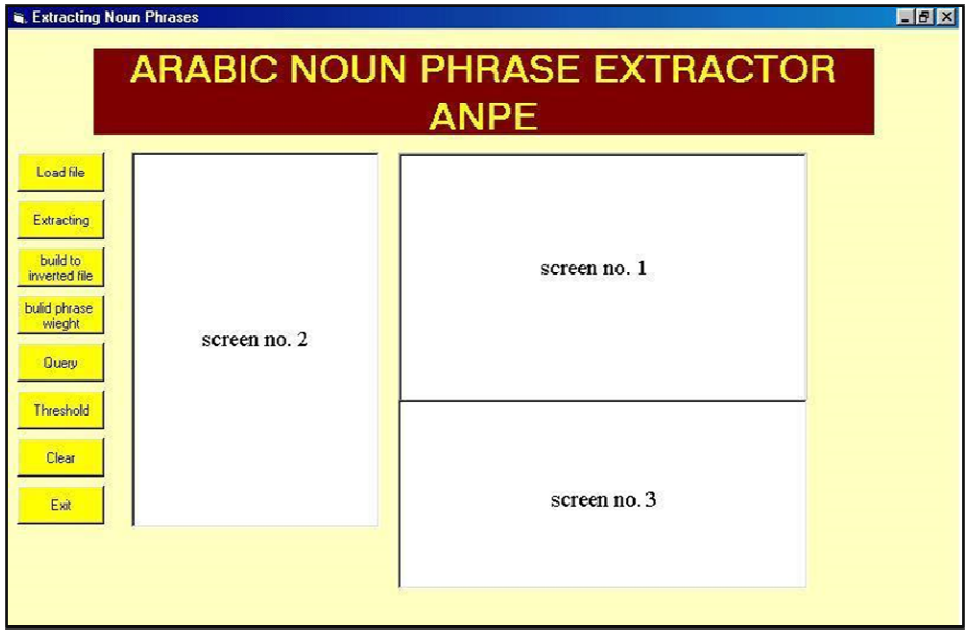


Figure 4

As noticed the general interface consists from screens (numbered as 1,2 and 3) and buttons (eight buttons).

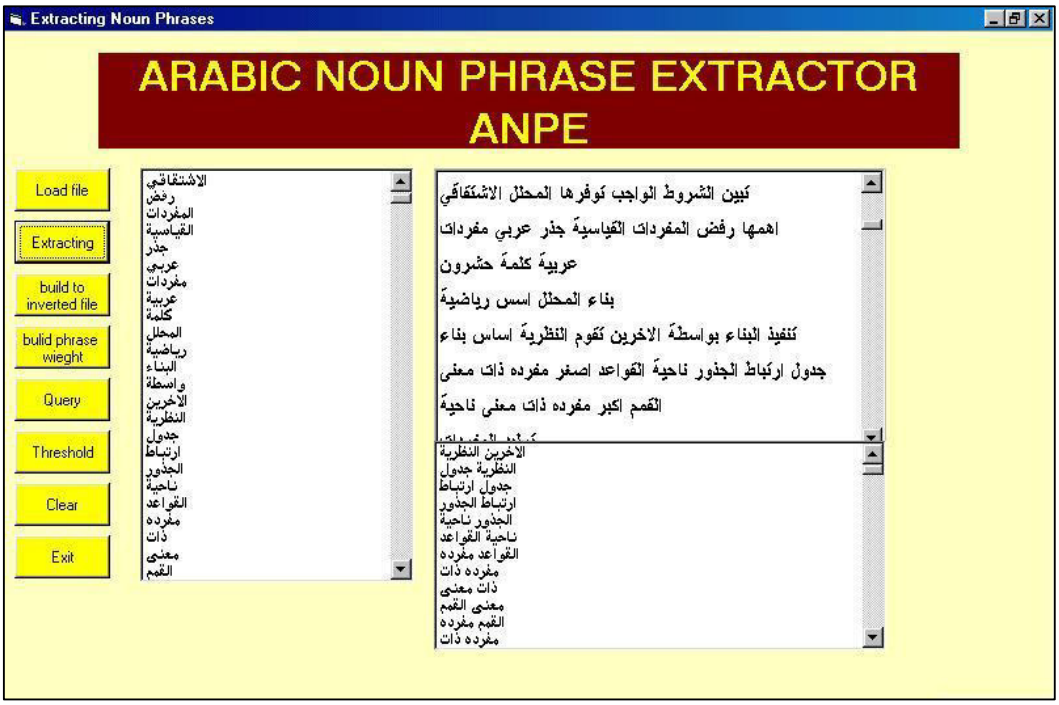
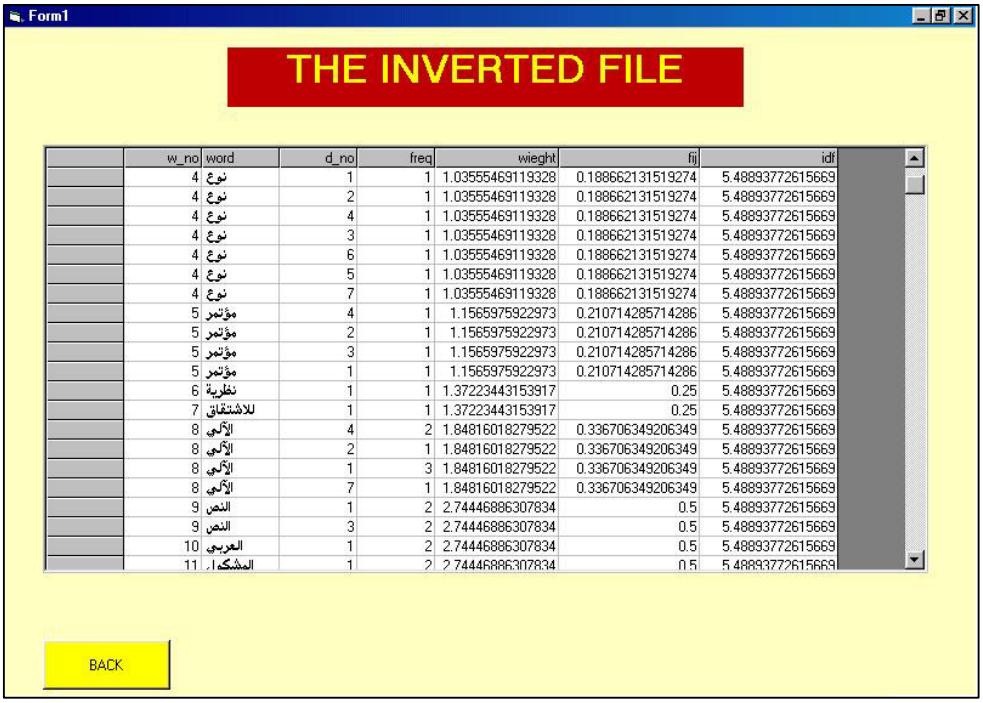


Figure 5

Figure 5 above illustrates the first three steps, which are opening the desired collection of documents and extract nouns from them then forming the phrases from the extracted nouns.
Here below Figure 6 illustrates the inverted file (mentioned in chapter five) built from a sample of seven documents:



	w_no	word	d_no	freq	wieght	fij	idf
	4	نوع	1	1	1.03555469119328	0.188662131519274	5.48893772615669
	4	نوع	2	1	1.03555469119328	0.188662131519274	5.48893772615669
	4	نوع	4	1	1.03555469119328	0.188662131519274	5.48893772615669
	4	نوع	3	1	1.03555469119328	0.188662131519274	5.48893772615669
	4	نوع	6	1	1.03555469119328	0.188662131519274	5.48893772615669
	4	نوع	5	1	1.03555469119328	0.188662131519274	5.48893772615669
	4	نوع	7	1	1.03555469119328	0.188662131519274	5.48893772615669
	5	مؤتمر	4	1	1.1565975922973	0.210714285714286	5.48893772615669
	5	مؤتمر	2	1	1.1565975922973	0.210714285714286	5.48893772615669
	5	مؤتمر	3	1	1.1565975922973	0.210714285714286	5.48893772615669
	5	مؤتمر	1	1	1.1565975922973	0.210714285714286	5.48893772615669
	6	نظرية	1	1	1.37223443153917	0.25	5.48893772615669
	7	للاشتقاق	1	1	1.37223443153917	0.25	5.48893772615669
	8	الآلي	4	2	1.84816018279522	0.336706349206349	5.48893772615669
	8	الآلي	2	1	1.84816018279522	0.336706349206349	5.48893772615669
	8	الآلي	1	3	1.84816018279522	0.336706349206349	5.48893772615669
	8	الآلي	7	1	1.84816018279522	0.336706349206349	5.48893772615669
	9	النص	1	2	2.74446886307834	0.5	5.48893772615669
	9	النص	3	2	2.74446886307834	0.5	5.48893772615669
	10	العربي	1	2	2.74446886307834	0.5	5.48893772615669
	11	المشكل	1	2	2.74446886307834	0.5	5.48893772615669

Figure 6

The system have been operated over a collection of 242 Arabic documents, the number of records in the inverted file reached 24091 records, and the constructed phrases were 32319 as shown in Figure 7.

Figure 7

After the user chooses the Query button appeared in Figure 4, Figure 8 will appear asking to enter the desired query.

Figure 9 illustrates the result of the query (الرياضيات) operated on the collection of the 242 documents.

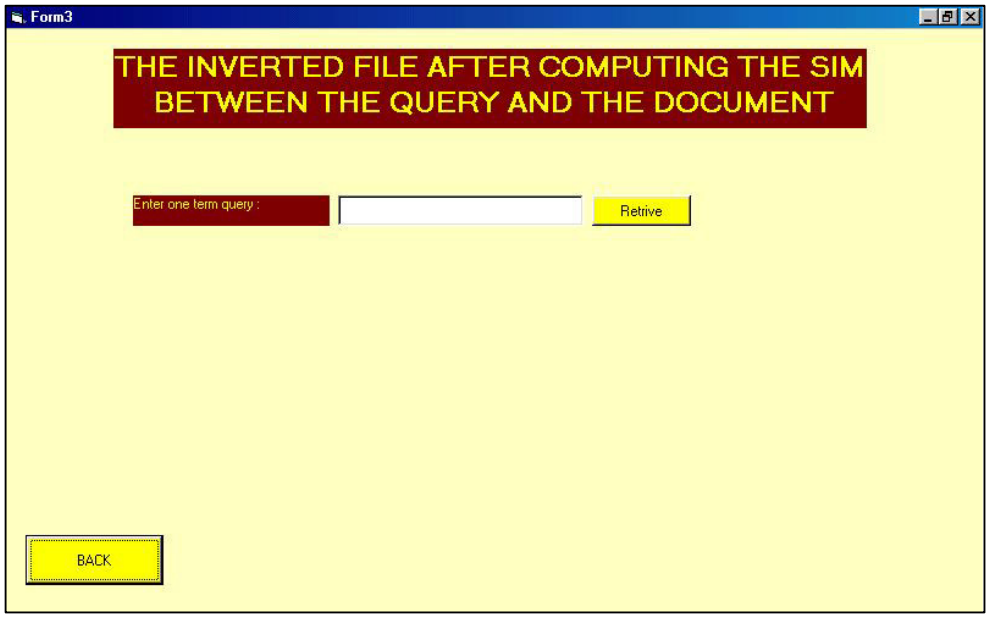
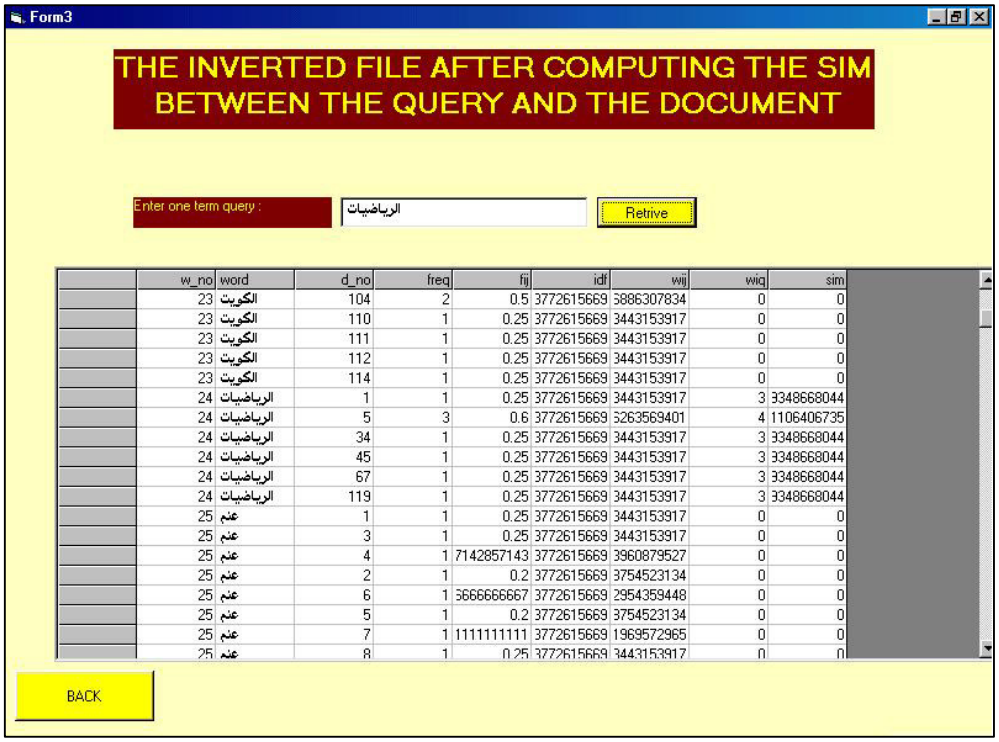


Figure. 8



	w_no	word	d_no	freq	fij	idf	wij	wiq	sim
	23	الكويت	104	2	0.5	3772615669	3886307834	0	0
	23	الكويت	110	1	0.25	3772615669	3443153917	0	0
	23	الكويت	111	1	0.25	3772615669	3443153917	0	0
	23	الكويت	112	1	0.25	3772615669	3443153917	0	0
	23	الكويت	114	1	0.25	3772615669	3443153917	0	0
	24	الرياضيات	1	1	0.25	3772615669	3443153917	3	3348668044
	24	الرياضيات	5	3	0.6	3772615669	3263569401	4	1106406735
	24	الرياضيات	34	1	0.25	3772615669	3443153917	3	3348668044
	24	الرياضيات	45	1	0.25	3772615669	3443153917	3	3348668044
	24	الرياضيات	67	1	0.25	3772615669	3443153917	3	3348668044
	24	الرياضيات	119	1	0.25	3772615669	3443153917	3	3348668044
	25	عنم	1	1	0.25	3772615669	3443153917	0	0
	25	عنم	3	1	0.25	3772615669	3443153917	0	0
	25	عنم	4	1	7142857143	3772615669	3960879527	0	0
	25	عنم	2	1	0.2	3772615669	3754523134	0	0
	25	عنم	6	1	3666666667	3772615669	2954359448	0	0
	25	عنم	5	1	0.2	3772615669	3754523134	0	0
	25	عنم	7	1	1111111111	3772615669	1969572965	0	0
	25	عنم	8	1	0.25	3772615669	3443153917	0	0

Figure 9

When choosing the Threshold, button appeared in Figure 4, Figure 10 will appear asking to enter the desired value for the threshold.

	phrase	wieght
	نصخ ورقية	2.85380687618996
	رقم البرمجة	1.42690343809498
	البرمجة فهر	0.7134517190474
	فهر نوع	2.83475135336778
	نوع مؤتمر	3.75635352684002
	مؤتمر عنو	3.13254262256295
	عنو لغة	2.95733485325974
	لغة برمجة	2.83455947493264
	برمجة عربية	3.55505149986906
	عربية علمي	4.52084821833323

Figure 10

Figure 11 illustrates the result of applying the threshold value (10) operated on the collection of the 242 documents, i.e. the extracted phrases that has weight values bigger or equal to 10 ranked decreasingly.

Figure 11

5. EVALUATION OF IR SYSTEM

To evaluate the performance of the IR system built in the previous section, we will use two popular retrieval evaluation measures: Recall & Precision.

The Recall is calculated according to the following formula:

$$Recall = \frac{No.of\ retrieved\ relevant\ documents}{No.of\ relevant\ documents}$$

The Precision is calculated according to the following formula:

$$Precision = \frac{No.of\ retrieved\ relevant\ documents}{No.of\ retrieved\ documents}$$

Here the two measures will be applied on both systems we obtained, the system which uses the single terms as an indexing unit (built in 6.1), and the system which uses the phrases (built in Section Four).

5.1 Results obtained by the System

The system had been operated over a collection of 242 Arabic documents; the phrases extracted reached 32319 phrases.

Figure.11 illustrates the Recall and Precision results, we notice that recall values for Noun Phrase are greater than it in Single Term, because using NPs retrieves more relevant documents, on the

other hand the precision values for Noun Phrase are smaller than it in Single Term, because the number of retrieved documents in NPs are less than it in Single Term.

Table 1: Recall & Precession Results

Query No.	Single term		Noun Phrase	
	Recall	Precision	Recall	Precision
2	7%	50%	67%	10%
3	7%	25%	79%	15%
4	20%	43%	60%	36%
8	57%	45%	73%	46%
9	7%	50%	100%	26%
13	21%	90%	36%	34%
16	33%	50%	33%	17%
17	75%	80%	81%	65%
18	33%	80%	88%	15%
19	4%	13%	21%	11%
20	67%	50%	83%	56%
21	20%	68%	91%	30%
22	38%	75%	75%	43%
23	50%	67%	93%	50%
27	7%	20%	36%	23%
32	14%	38%	86%	20%
35	4%	100%	43%	54%
39	21%	50%	50%	20%
40	14%	13%	53%	21%
49	30%	33%	60%	15%
50	12%	21%	64%	15%
57	45%	64%	65%	22%
59	3%	100%	70%	40%

6. CONCLUSIONS

This paper introduced the Arabic Noun Phrase Extractor system, three common steps were used to build it: tokenization; part-of-speech tagging; and noun phrase identification.

At the end of the system Run, a list of Noun Phrases was included in the documents, sorted in decreasing order according to their weights in the documents, obviously, the phrases on the top of the list – i.e. has large weights – are said to be the best to represent the documents' content
The experiment has been executed using 242 documents, the extracted phrases were nearly 32319 noun phrases, 23 queries were used for both single terms and NPs, the results proved that when using phrases for indexing the Recall value becomes better than using single terms, i.e. the system yields more relevant documents from the retrieved ones, on the other side it gave low precision because number of the retrieved documents will be decreased.

The is beginning of an ongoing research, there must be more efforts to include more intelligent ways for analysing data, it can be enhanced by many effective Machine Learning approaches

such as Hidden Markov Model. further ANPE was built using VB6, it will be more efficient if it is built using Natural Language Toolkit (NLTK), which was written in Python,

REFERENCES

[1] Jonathan, Owens (2013). The Oxford Handbook of Arabic Linguistics. Oxford University Press. p. 2. ISBN 0199344094. Retrieved January 27, 2017.

[2] Internet World Stats. (2017). Internet world stats usage and population statistics. Retrieved January 27, 2017, from http://www.internetworldstats.com/

[3] El Younoussi Yacine., (2015). Towards an Arabic Web-based Information Retrieval System (ARABIRS): Stemming to Indexing. International Journal of Computer Applications 109(14):28-33,

[4] Lujain Alkhazy, (2016). Noun Phrases in Arabic: a Descriptive Study of Noun Phrases in Modern Standard Arabic and Najdi Arabic, Master Thesis in California State University, Northridge.

[5] Islah Gharaibeh & Natheer Gharaibeh, (2012), Towards Arabic Noun Phrase Extractor (ANPE) Using Information Retrieval Techniques, in Journal of Software Engineering, Scientific & Academic Publishing , Volume 2, Number 2 , pp 36-42 ,

[6] Saleem Abuleil-Khalid Alsamara-Martha Evens, (2002), Acquisition system for Arabic noun morphology, - Proceedings of the ACL-02 workshop on Computational approaches to semitic languages.

[7] Eman Othman, Khaled Shaalan, Ahmed Rafea, (2003) "A Chart Parser for Analyzing Modern Standard Arabic Sentence", Cairo Univ.,

[8] Lahtinen, T., (2000), Automatic indexing: An approach using an index term corpus and combining linguistic and statistical methods. Report, University of Helsinki, Helsinki

[9] Vishwa Vinay, (2003), Automatic Key Phrase Assignment, Master Thesis in University College London.

[10] White paper (2001), "Enabling E-business in Multiple Languages", Verity Internationalization,

[11] Bennett N. A., He Q., Powell K., Schatz B. R. (1999), Extracting noun phrases for all of MEDLINE. Proceedings of the AMIA Symposium; Washington, DC, USA. American Medical Informatics Association; pp. 671–675

[12] Bennett, N. A., He, Q., Chang, C. T. K., & Schatz, B. R. (1999), "Concept Extraction in the Interspace Prototype", Technical report Dept. of Computer Science, University of Illinois at Urbana-Champaign, 1999.

[13] James Allen. (1995), Natural Language Understanding, The Benjamin / Cummings Publishing Company, Inc., 2nd Edition

[14] Gerard Salton, (1988) "Syntactic Approaches to Automatic Book Indexing", 88 Proceedings of the 26th annual meeting on Association for Computational Linguistics Pages 204-210