International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019

Keyword Extraction from Arabic Text using the Page Rank Algorithm

Meran M. Al Hadidi, Muath Alzghool, Hasan Muaidi

***Abstract:** This paper describes how keywords are extracted from Arabic text using the page rank algorithm, by constructing a graph whose vertices are formed by candidate words that are extracted from the title and the abstract of a given Arabic text after applying a tagging filter to that text. Next, a co-occurrence relation is applied to draw the edges between the vertices within specified window sizes. Then, the page rank algorithm is applied to the graph to rank the importance of each keyword. Finally, the vertices are sorted in descending order by their page rank scores and the tokens with highest scores are chosen as the keywords. Several experiments were conducted on a dataset that consisted of 100 Arabic academic articles for training and 50 for testing. The results were evaluated by using precision, recall, and the F-measure. The maximum recall achieved on the dataset was 63%, as not all the manually identified keywords and keyphrases existed in the article abstracts and titles. The proposed method achieved 25% of recall, which is acceptable as it is comparable to that of a method in the literature that was applied to an English language testing dataset that consisted of 500 English documents, which achieved 42% of recall where the maximum recall percentage of the testing dataset was 78%. Despite the difficulties and challenges in searching for keywords in the Arabic language and using fewer documents in the Arabic testing dataset than in the English, it can be concluded that the proposed keyword and keyphrase extraction system using the page rank algorithm works well.*

***Keywords:** Text Processing, Arabic Language, TextRank Algorithm, Keyword Extraction*

I. INTRODUCTION

Keywords and keyphrases can be defined as a sequence of one or more words extracted from a document that provide important and descriptive information about the content of a document, and such words and phrases may serve as either an indicative summary or as document metadata, both which can help the reader in their search for relevant information. Keywords can be used to facilitate the more efficient searching of information and to help the reader to make a decision about whether it is necessary to read a document [11]. They can be useful in various applications such as retrieval engines, browsing interfaces, thesaurus construction or text mining [12].

Revised Manuscript Received on October 30, 2019.
* Correspondence Author

Meran M. Al Hadidi*, Instructor at Al-Balqa Applied University in Princess Alia University College, Shmeisani, Amman, Jordan.
Dr. Muath Alzghool, M.Sc. Degrees in computer science from Yarmouk University, Jordan.
Dr. Hasan Muaidi, Associate Professor at Al-Balqa Applied University, in Faculty of Huson College, Irbid, Jordan.

The manual selection of keywords and keyphrases from a document by a human is an extremely slow, difficult, and time-consuming process that can lead to errors, so it is almost impossible to extract keywords successfully in this manner. Hence, automatic keyword and keyphrase extraction is an important task that needs to be automated due to the benefits it can offer in managing information [13]. Automatic keyword extraction should be done systematically and with either minimal or no human intervention, depending on the model. The goal of automatic extraction is to apply the power and speed of computation to the problems of access and discovery. Moreover, it obviates the significant cost and difficulties associated with the use of human indexers in the task of information retrieval [5]. Many keyword extraction algorithms have been developed and implemented in recent years. Perhaps one of the most notable is the keyword extraction system using the page rank algorithm, which was implemented on English-language text by Mihalcea and Tarau (2004) [8]. For their system, they introduced the text rank, a graph-based ranking model for text processing, and showed how this model can be utilized successfully in natural language processing (NLP) applications, and especially in the extraction of keywords. The subsequent keyword extraction system proposed by Mihalcea, Liu, and Lieberman (2006) [9] is another tried and tested in-formation retrieval application for NLP and can extract the required information from a large database. (Natural language processing can be defined as the ability of a computer program to understand human speech as it is spoken.) [14]. However, these studies, like the majority of research in this area, were conducted on texts in the English language. In contrast, very few research studies have been carried out on methods to facilitate the key- word searching of Arabic-language text. This may because, compared to the English language, Arabic is more challenging to process. Arabic is a rich language with a complex morphology; it has very different and difficult structures compared to other languages. It consists of 28 letters written from right to left. Text is written, in a cursive style, which means that Arabic can- not be written as unconnected, separated letters as in English. In other words, in general, all the letters must be connected together. Moreover, there are no capitals or small letters in Arabic, and letters can be written in four different forms depending on their location in a word. In fact, most of the letters in Arabic have four forms: Stand-alone, word initial, medial and word ending depending on their position in a word [4]. In addition, Arabic words have two genders, feminine and masculine; three numbers, singular, dual, and plural; cases, nominative, accusative, and genitive [3]. As explained by Namly, Bouzoubaa, Tahir, and Khamar (2015) [10], Arabic words can be divided into three types: Noun, verb and particle, the authors state that a noun is a token that has a meaning in itself without needing to be connected with time. A verb is a token that indicates a state or a fact happening in the past, present, or future. Particles are used in sentence construction [1]. A particle may consist of more than one letter and can be used to convey the following types of meaning: introduction, exclusion, restriction, inauguration, interrogation, future, rectification, imperative, stimulation, authenticity, selection, solicitation, similitude, variability, astonishment, definition, causality, interpretation, separation, paucity, profusion, wish, premonition, regret, confirmation, answer, rejection, augmentation, condition, circumstance, exposition, attraction, finality, oath, originality, surprise, lamentation, call, negation, and interdiction. Furthermore, the Arabic language is a Semitic language that is characterized by a wide number of linguistic variants. The characteristic feature of Semitic languages is their basis in consonantal roots, which are mostly trilateral (three-lettered). In order to generate Arabic text, a proper understanding of Arabic grammar is needed. Arabic grammar consists of morphology and syntax. Morphology refers to the forms of words and their transformations into intended meanings. Syntax refers to the case endings of words and their positions in the sentence. From the above, it is clear why there are several challenges to be over- come for the successful extraction of keywords from Arabic text. The complexity of the language may also be the reason why transliteration into another language such as English can be difficult. Transliteration can be defined as a method of mapping the script of one language to the best matching script of another language, word by word, or perfectly letter by letter. It is used to represent Arabic words for readers who cannot read the Arabic script. Compared to the large number of publications and available resources and lexicons in English, there are fewer publications and very few resources for Arabic. Thus the lack of resources, the variant sources of ambiguity, and rich metaphoric script usage remain the most challenging problems for Arabic NLP researchers. In sum, the Arabic language is a highly inflected language; it has much richer morphology than English. Hence the need for more studies in this area and the motivation for this research which seeks to apply the page rank method often used for English texts to the problem of keyword extraction from Arabic texts.

II. PAGE RANK ALGORITHM

Perhaps the most famous keyword search engine is Google, which is owned by Google, Inc. The company states that its mission is to organize the world's information and make it universally accessible and useful. Google is the largest search engine on the web; it receives over 200 million queries each day through its various services. The page rank approach lies at the heart of the Google's search software. The page rank algorithm for ranking web pages was developed by Larry Page and Sergey Brinat Stanford University in the late 1990s as a means, to determine how much more important a certain web page is compared to other web pages. For their method, they use a common type of authority measure, namely, the in-degree measure for a specific page [2]. In essence, the page rank algorithm is applied to a graph whose vertices represent individual web pages, in order to determine the importance of a certain vertex (a web page) within this graph. It does this by taking into account global information recursively computed from the entire graph, rather than by relying only on local vertex-specific information. The basic concept that is applied in a graph-based ranking model is that of voting. When one vertex (web page) links to another one, it is basically creating a vote for that other vertex. Thus, the higher the number of votes for a vertex; the higher the importance of that vertex. Each vertex is given a score that is determined by the number of votes that are cast for by other vertices, as well as the scores of the vertices that have cast these votes [8]. So let G (V, E) refer to the constructed graph, where V is the set of vertices in the graph that represents the web pages, and E is the set of edges between the vertices. An edge is constructed in the graph between two vertices when one vertex points to another vertex. So, for a given vertex V, let In (V) represents the set of vertices that point to the vertex (V), and Out (V) represent the set of vertices that points to. Thus, according to Equation 1 in [2], the score of a given vertex (Vi) is computed as follows:

$$S(Vi) = (1-d) + d * \sum_{j \in In\,(Vi)} \frac{1}{Out(Vj)}\ S(Vj) \qquad (1)$$

Where d refers to the damping factor whose, value can be set between 0 and 1. This role of this factor is to integrate into the model the probability of jumping from a given vertex to another random vertex in the graph. The factor value of d is usually set to 0.85 [2]. Therefore, this value is used in the current research implementation.

III. KEYWORD EXTRACTION USING THE PAGE RANK ALGORITHM

In the method for keyword extraction outlined in Mihalcea and Tarau (2004) [8], a net- work graph is constructed using candidate keywords as nodes, where co-occurrence is used to draw edges between them, and then the page rank algorithm is applied to the graph to rank the importance of each keyword. Their text rank algorithm makes use of the Hulth (2003) dataset. This dataset consists of 2000 English abstracts from the international Information Science, Physical Sciences, Engineering and Computer Sciences (INSPEC) database from the years 1998 to 2002 and includes articles Computers and control, and information technology (IT). Keywords were assigned by a professional indexer into two sets: A set restricted to terms in the INSPEC Thesaurus, and an uncontrolled set. It should be noted that in their work, keywords were assigned from the full documents; not just the abstracts. The resulting keyword dataset was divided: 1000 for training, 500 for validation, and 500 for a hold out test.

A. Dataset

In line with the above approach, in the current research, the text rank algorithm was tested on a collection of articles published in the Arabic language that were collected manually from the Internet from a range of disciplines: Islamic law, basic and social sciences, child-rearing and IT. This dataset was divided into two sets: 100 documents for training, and 50 documents for a hold-out test set. Some statistics were calculated for this dataset which was of great benefit in the implementation of experiment. The Arabic abstracts with their titles were reprinted in Notepad files.

Also, their corresponding keywords were also reprinted in Notepad files with the same names as the abstract files but with different extensions. Every file had its own total number of keywords and keyphrases that were built manually. Table I shows the number of keyphrases that were found in the abstract files, which ranged from 2 to 12, the table also shows the total frequency of abstract files that contained the specified numbers of keyphrases.

Table I. Number and Frequency of Keyphrases in the Dataset

Number of Keyphrases	Total No. of Documents
2	8
3	63
4	41
5	23
6	7
8	6
10	1
12	1

From Table I, it can be seen that the minimum number of keyphrases that were found in a document was 2, and the maximum number was 12. Table II summarizes the information in Table I as a computed average number of keywords and keyphrases that can be found in a document. Note that the keyphrases consist of one word or more. In order to determine the length of each keyphrase, a full statistical analysis was performed on these keyphrases.

Table II. Minimum, Maximum and Average Number of Keywords and Keyphrases in the Dataset

Minimum Number of Keyphrases	2
Maximum Number of Keyphrases	12
Average Number of Keyphrases per Document	3.973333333

The keyphrases were identified for four different situations (search locations):

1. Document title
2. Document abstract
3. Document title or abstract
4. Document title and abstract

Table III shows the number of existing and non-existing keyphrases for the above mentioned four different search locations. Statistics in Table III are a great benefit in the weighted PageRank equation. This is because the search results can be improved by using suitable weights in the PageRank equation. For example, in this research, the keywords and keyphrases that were automatically found in titles were given more weight; because it was observed that most of the keyphrases in the abstracts also existed in their respective titles, The table shows the maximum expected recall for each situation (search location), which was calculated by dividing the total existing keywords and keyphrases by the total number of keywords and keyphrases (596) that were found manually in the dataset. It shows that the maximum recall that can be achieved when searching the documents is 63% (abstracts or titles). In other words, the ideal recall percentage that can be achieved is 63% not 100% because not all the keywords and keyphrases existed in both the abstracts and titles of all the documents. It can be seen from Table III that based on the title or abstract (document) search location results, the minimum keyphrase length (in words) is one and the maximum is six.

Table III. Number of Keyphrases with Different Lengths (in Words) from Different Search Locations

Search Loca -tion	1 Word	2 Word s	3 Word s	4 Word s	5 Word s	6 Word s	Recall in %
Title	61	167	33	7	3	0	45.46
Abstract	80	192	38	8	3	1	54.02
Title or Abstract	90	227	46	9	3	1	63.08
Title and Abstract	51	132	25	6	3	0	36.40

The following steps were used to determine the number of retrieved keyphrases based on their length:

1. Calculate the total existing keyphrases in the abstracts and their titles (376 out a total of 596).
2. Determine the percentage of retrieved keywords from the total words in each document (30%).
3. Determine the number of retrieved keyphrases (12; because the maximum number of keyphrases in a document in the dataset is 12, as shown in Table I).
4. Focus on the keyphrases that consist of one, two and three words because they constitute 96.54% of the total existing keyphrases.
5. Determine the number of one-word (Single-Word) keyphrases according to the following equation:

$$12 * \frac{90}{376} = 3$$

Where 12 is the total number of keyphrases retrieved, 90 is the number of existing one-word keyphrases in the documents and 376 is the total number of existing keyphrases in the documents.

6. Determine the number of two-word keyphrases according to the following equation:

$$12 * \frac{227}{376} = 7$$

Where 227 is the number of existing two-word keyphrases in the documents.

7. Determine the number of three-word keyphrases according to the following equation:

$$12 * \frac{46}{376} = 2$$

Where 46 is the number of existing three-word keyphrases in the documents.

From the above method, the number of automatic keyphrases that is retrieved is 12, which is divided as follows into: Three one-word keyphrases, seven two-word keyphrases and two three-word keyphrases.

B. *Research Methodology*

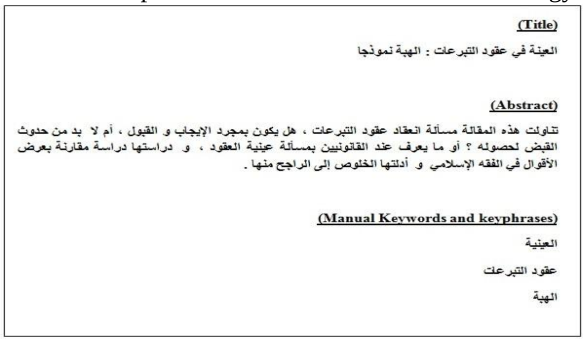
The algorithm of the proposed system consists of the following main steps:
Input: A dataset of Arabic documents (abstracts and their titles).
Output: A set of keywords and keyphrases for each Arabic document (abstract with its title).
Begin

- Boundaries of Arabic sentences are detected and each sentence is separated.
- Arabic words in each sentence are tokenized and tagged with three main part-of-speech (POS) tags: Verb (V), Noun (N, N1 and NT) or Particle (P).
- Arabic nouns are divided into three main types:
  - (N) refers to general nouns that are not attached to any type of particle
  - (N1) refers to nouns that are attached to particles such as pronouns or attributes that describe these nouns
  - (NT) refers to general nouns that are not attached to any type of particle and appear in document titles.
- Punctuation marks between words in each document are tokenized as punctuation marks that separate sentences in files.
- Vertices are chosen from the text based on manual tagging, where only nouns of types (N and NT) are selected. According to the training dataset, user tends to use nouns as keywords in the Arabic language. This differs from the situation for the English language, where users tend to use nouns and adjectives as keywords.
- Edges are drawn between vertices that fall within a co-occurrence window of size n, and these edges can be forward, backward or bidirectional.
- The page rank algorithm is applied on the constructed graph using an initial value of 1 for each vertex until convergence to within a specified threshold value occurs [6].
- Vertices are sorted by their page rank scores in descending order and a percentage of the total number of tokens in each document is chosen as keywords. Each token is expanded into a set of keyphrases by searching for each occurrence of the token in the original text, and for each occurrence all adjacent words that are eligible tokens and collected and concatenated into a phrase [8].

In this research, the top 30% of the total number of tokens in each document are selected as keywords according to their page rank scores. For example, if the total number of words in a document is 100, then just the top 30 vertices are selected as keywords. While, if the total number of words in a document is 85, a ceil function is used to ignore the fractional part and round up to the next integer, then the top 26 vertices are selected as keywords.

C. *Creation of Graphs for Arabic Texts*

There are two main steps in the methodology adopted for this research: (1) construct a graph from an Arabic text and, (2) application of the page rank algorithm to the constructed graph. Fig. 1 shows an example of an Arabic document from the collected dataset. The document is an abstract and it is named (20.ABSTR) and is used as to illustrate the steps followed in the research methodology.



(Title)

العينة في عقود التبرعات : الهبة نموذجا

(Abstract)

تناولت هذه المقالة مسألة انعقاد عقود التبرعات ، هل يكون بمجرد الإيجاب و القبول ، أم لا بد من حدوث القبض لحصوله ؟ أو ما يعرف عند القانونيين بمسألة عينية العقود ، و دراستها دراسة مقارنة بعرض الأقوال في الفقه الإسلامي و أدلتها الخلوص إلى الراجح منها .

(Manual Keywords and keyphrases)

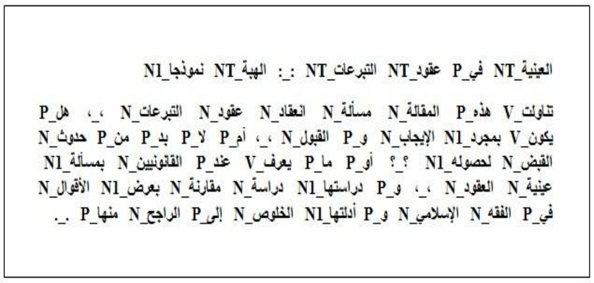
العينية

عقود التبرعات

الهبة

Fig. 1. Example of Arabic abstract document (20.ABSTR).

First, every token in the document is tokenized and tagged, as shown in Fig. 2.

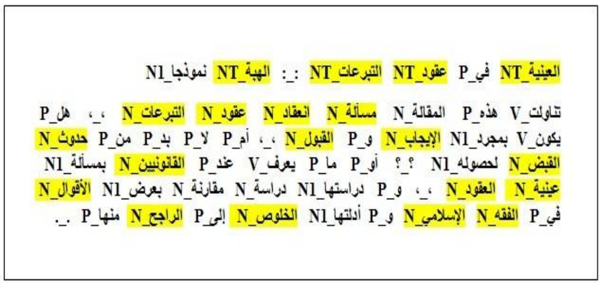


العينية_NT في_P عقود_NT التبرعات_NT :_: الهبة_NT نموذجا_N1

تناولت_V هذه_P المقالة_N مسألة_N انعقاد_N عقود_N التبرعات_N ،_، هل_P
يكون_V بمجرد_N1 الإيجاب_N و_P القبول_N ،_، أم_P لا_P بد_P من_P حدوث_N
القبض_N لحصوله_N1 ؟_؟ أو_P ما_P يعرف_V عند_P القانونيين_N بمسألة_N1
عينية_N العقود_N ،_، و_P دراستها_N1 دراسة_N مقارنة_N بعرض_N1 الأقوال_N
في_P الفقه_N الإسلامي_N و_P أدلتها_N1 الخلوص_N إلى_P الراجح_N منها_P ._.

Fig. 2. Tokens and Part-of-Speech Tags for document (20.ABSTR).

Then, after the stopwords have been removed, the graph of the document is constructed by using nouns of types (N and NT) as vertices. The co-occurrence relation is used to connect the selected vertices, which is controlled by the distance between the word occurrences. All the lexical units that pass the above syntactic filter are added to the graph, and an edge is added between those lexical units that co-occur within a window of N words, where N can be set anywhere from between 2 to 10 words. In Fig. 3 the tokens shaded in yellow are the vertices for the document (20.ABSTR) that are selected to construct a graph after the data have been filtered and only the tokens of types N or NT have been selected. The right part of the shaded text represents the token (word) and the left part represents its POS tag; note that the two parts are separated by an underscore (_).


العينية_NT في_P عقود_NT التبرعات_NT :_: الهبة_NT نموذجا_N1

تناولت_V هذه_P المقالة_N مسألة_N انعقاد_N عقود_N التبرعات_N ،_، هل_P
يكون_V بمجرد_N1 الإيجاب_N و_P القبول_N ،_، أم_P لا_P بد_P من_P حدوث_N
القبض_N لحصوله_N1 ؟_؟ أو_P ما_P يعرف_V عند_P القانونيين_N بمسألة_N1
عينية_N العقود_N ،_، و_P دراستها_N1 دراسة_N مقارنة_N بعرض_N1 الأقوال_N
في_P الفقه_N الإسلامي_N و_P أدلتها_N1 الخلوص_N إلى_P الراجح_N منها_P ._.

Fig. 3. Vertices selected from document (20.ABSTR).

Edges in a graph may be directed or bidirectional. A bidirectional graph is a graph in which the vertices or nodes are connected together, where all the edges are bidirectional. In contrast, a graph in which the edges point in a certain direction is called a directed graph. A directed graph may be a forward or backward graph. In a forward graph, vertices are connected together, where each vertex points to its successors based on a certain relation, while in a backward graph, all the vertices are connected but each vertex points to its ancestors, also according to a specific relation [7]. In the first experiment, the above three graph types (bidirectional, forward, back- ward) are implemented separately on the training dataset. Then other features are added to each graph type gradually to determine whether those features improve the outcomes. The results are compared based on the highest average F-measure value in order to select the best settings. These settings are then applied to the test set.

D. Application of the Graph Types to the Documents

Bidirectional Graphs Each edge in a bidirectional graph has two directions. When the window size is set to 3, the tokens shaded yellow (vertices of the constructed graph) in Fig. 3 are tested to determine whether these vertices exist within a window size of 3 in the original document. If they do, then there is a bidirectional relation between these vertices according to their locations in the original document. Fig. 4 shows the bidirectional graph built for document (20.ABSTR) with a window size of 3.

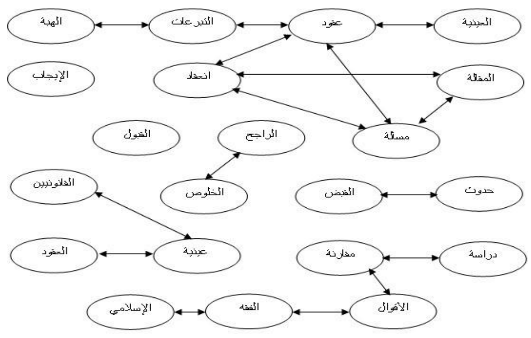


Fig. 4. Bidirectional graph for document (20.ABSTR) with window size of 3.

Forward Graphs In a forward graph every vertex points to its successors according to the used window size and its position in the text. Fig. 5 shows the forward graph built for document (20.ABSTR) with a window size of 3.

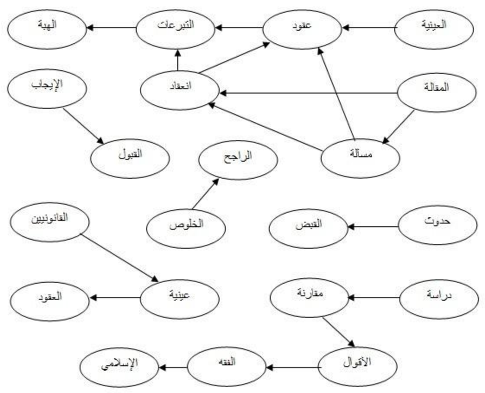


Fig. 5. Forward graph for document (20.ABSTR) with window size of 3.

Backward Graphs In contrast to the forward graph, in the backward graph, every vertex points to its ancestors, also based on the window size and its position in the text. Fig. 6 shows the forward graph built for document (20.ABSTR) with a window size of 3.

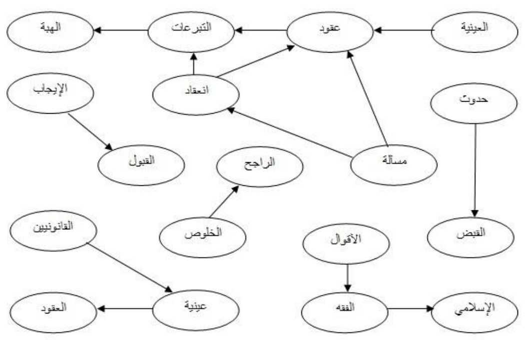


Fig. 6. Backward graph for document (20.ABSTR) with window size of 3.

E. Applying the Page Rank Algorithm

After the vertices are added to the constructed graph they are restricted through the use of syntactic filters that select only the lexical units of a certain part of speech, such as only nouns of type (N or NT), for each Arabic document (abstract and its title) in the sample. All the lexical units that pass the syntactic filter are added to the graph, and then edges are drawn between those lexical units that co-occur within a predefined word window. After the graph is constructed (bidirectional, forward or backward), the score associated with each vertex is set to an initial value of 1. Then the page rank algorithm (Equation 1) is run on the graph vertices for several iterations until it converges. This process usually requires 20 to 30 iterations to reach an error threshold of 0.0001. Convergence is achieved when the error rate for any vertex in the graph falls below that threshold. The error rate is approximated by using the difference between two successive scores for each of the graph vertices. When a final score has been obtained for each vertex of the graph of a document, the vertices are sorted in reverse order according to their page rank scores. Then the top 30% of the total number of tokens that exist in a document are retrieved for use in the post-processing steps for keyword and keyphrase extraction into a page rank list.

Note that the value of 30% was selected after comparing the results of several experiments conducted on the training test to identify the best possible rate that would lead to the best results, as shown in Equation 2.

$$\boldsymbol{Keywords = |30\,percent * Total\,Tokens|} \quad (2)$$

Where Total Tokens refers to the total number of tokens in the document. This number is equal to 55 in document (20.ABSTR). Therefore, according to Equation 2 above, the number of selected vertices =17.

$$\boldsymbol{Keywords = |30\,percent * 55|}$$

$$\boldsymbol{Keywords = |16.5|}$$

$$\boldsymbol{Keywords = 17}$$

Hence, the top 30 percent of the words (17 words) extracted from document (20.ABSTR) in Fig. 3 are retrieved for the post-processing stage in this system.

F. *Post-Processing Steps to Extract Keywords and Keyphrases*

After removing stopwords, applying the above syntactic filters that select only lexical units of a certain part of speech, assigning a page rank score for each vertex in the constructed graph, arranging the vertices in descending order according to their page rank scores and choosing the top 30% of the total number of words in the document as keywords, the post-processing stage to extract keywords and keyphrases takes place as follows:

Input:
A list of Arabic documents (abstracts and their titles), and a list of selected keywords for each document.
Output: A set of keywords and keyphrases for each Arabic document (abstract with its title).
Begin

1. Scan the original document token by token.
2. Add all single tokens found in the text and apply the page rank algorithm to return a candidate list of keyphrases with the corresponding page rank score.
3. Add all successive tokens found in the original text as a keyphrase to the keyphrase list with the corresponding page rank score which is equal to the summation the of tokens' page rank scores constituting that keyphrase.
4. Ignore all keyphrases of size greater or equal to four tokens.
5. Remove all keyphrases of size 3 (i.e, those that consists of three tokens) if they have a candidate sub-keyphrase of size 2 (i.e., consisting of two tokens).
6. Sort all the returned keywords and keyphrases based on their page rank scores, then:
   - Add the top three keyphrases of size 1 (one word) to the final list of keyphrases.
   - Add the top seven keyphrases of size 2 (two words) to the final list of keyphrases.
   - Add the top two keyphrases of size 3 (three words) to the final list of keyphrases.

To summarize, all the tokens in the original document are scanned and if there are two or more successive keywords then a keyphrase is composed out of these successive keywords. Else a keyword consisting of a single word is selected. Thus, keywords and keyphrases of several lengths are composed. Based on an assessment the statistics derived from the used dataset, it was found that the percentage of actual keyphrases with lengths of one, two and three words constituted the greatest proportion of the total, at about 96.54%. Therefore, automatic keyphrases containing one, two and three words were selected and automatic keyphrases with lengths greater than three words were ignored. Finally, a specific number of retrieved automatic keywords and keyphrases of two or three words in length are selected according to some statistics derived from the used data. Hence, for document (20.ABSTR), The top three keywords of one-word in length, the top seven keyphrases of two words in length and the top two keyphrases of three words in length were selected as the extracted keywords and keyphrases.

G. *Weighted Vertices in Graphs*

Two methods are used to apply weights to the graph vertices:

- The final page rank score for each token is multiplied by its frequency in the document.
- Each page rank score for the selected keyword that exists in the document title is multiplied by a weight value that is determined by (3), after applying several experiments.

The results of the experiment showed that using weights for the graph vertices improved the results significantly. Vertices are connected through links (edges). These links may be strong or poor depending on several factors. For example, words may occur more than once in a document. In this study, weights are applied to the vertices in the constructed graph according to frequency of the term (word) in the document and the positions of the term (word) with in the document, either in the title or the abstract.

IV. EXPERIMENTS RESULTS

The following is a list of additional features and techniques that were employed in the proposed system in order to determine whether and to what extent their usage would improve the results of the page rank algorithm:

1. Graph Type.
2. Removal of stopwords.
3. POS tagging.
4. Window size (2, 3, 5 or 10).
5. Weights for vertices.

The results of applying the above features are presented in the following subsections.

A. *Experiment 1: Effect of Graph Type*

The first experiment was performed to determine the effect of the graph edge direction on the effectiveness of the proposed keyphrase extraction system using the page rank algorithm. The texts in this experiment include stopwords but POS tagging was not applied to the tokens in the documents.

The texts were taken from the training dataset. Tables IV, V and VI show the results of applying the proposed system using a bidirectional graph, forward graph and backward graph, respectively, with window sizes of 2, 3, 5 and 10 on the training dataset.

Table IV: TextRank Using Bidirectional Graph (BI) and Various Window Sizes (w) on the Training Dataset

Experi-ment Name	Rele-vants	Ret-rieved	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measure
BI, w=2	401	1142	12	0.0352	0.0110	0.0162
BI, w=3	401	1142	12	0.0352	0.0110	0.0162
BI, w=5	401	1142	12	0.0352	0.0110	0.0162
BI, w=10	401	1142	12	0.0352	0.0110	0.0162

Table V: TextRank Using Forward Graph (FD) Various Window Sizes (w) on the Training dataset

Experi-ment Name	Rele-vants	Ret-rieved	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measure
FD, w=2	401	1074	8	0.0196	0.0071	0.0101
FD, w=3	401	1079	9	0.0236	0.0089	0.0126
FD, w=5	401	1073	10	0.0240	0.0114	0.0145
FD,w=10	401	1043	15	0.0363	0.0180	0.0209

Table VI: TextRank Using Backward Graph (BD) Various Window Sizes (w) on the Training Dataset

Experi-ment Name	Rele-vants	Ret-rieved	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measure
BD, w=2	401	1095	2	0.0058	0.0017	0.0027
BD, w=3	401	1088	2	0.0050	0.0019	0.0028
BD, w=5	401	1078	7	0.0203	0.0067	0.0100
BD,w=10	401	1051	10	0.0248	0.0119	0.0146

Based on the above results, it can be seen that the forward graph has the highest F-measure values in all window sizes. Therefore the forward graph was adopted instead of the backward and bidirectional graphs and the rest of experiments were performed using the forward graph only. Fig. 7 contains a bar chart of the results of the above experiments using window sizes of 2, 3, 5 and 10 and bidirectional (BI), forward (FD) and backward (BD) graphs on the training dataset.

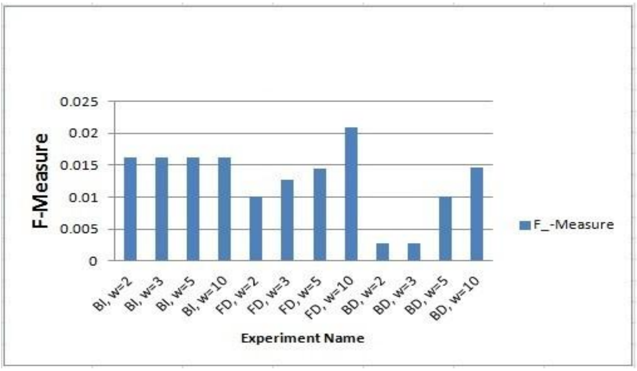


Fig. 7. F-measure Using Bidirectional (BI), Forward (FD) and Backward (BD) graphs with Different Window Sizes on Training Dataset.

B. *Experiment 2: Effect of Stopword Removal*

The second experiment involved removing the stopwords from the training dataset and then applying the proposed system using the forward graph, to window sizes 2, 3, 5 and 10 but without using the POS tagging process. Table VII shows the results. As can be seen from the table, the removal of stopwords from the documents as a preprocessing step before applying the proposed system leads to better results. Since stopwords are very common words that appear frequently in text they carry little meaning; they serve a syntactic function but do not indicate subject matter. Their presence can cause problems when searching for phrases that can best describe a document, so their removal tends to yield better results because the competition for a page rank is then just between the required tokens. The effectiveness of stopword removal is graphically illustrated by Fig. 8 which shows a comparison of the F- measure values for different window sizes of 2, 3, 5 and 10 using the forward graph on the training dataset with and without stopwords.

Table VII: TextRank Using Forward Graph (FD) and Various Window Sizes (w) on the Training Dataset after Removal of Stopwords

Experi-ment Name	Rele-vants	Ret-rieved	Ret-rieved Rele-vants	Re call	Pre-cesion	F-Meas ure
FD, w=2	401	1141	47	0. 12 59	0.0396	0.058 9
FD, w=3	401	1131	53	0. 14 64	0.0456	0.068 2
FD, w=5	401	1139	58	0. 16 05	0.0519	0.075 4
FD, w=10	401	1135	63	0. 17 50	0.0566	0.08 27

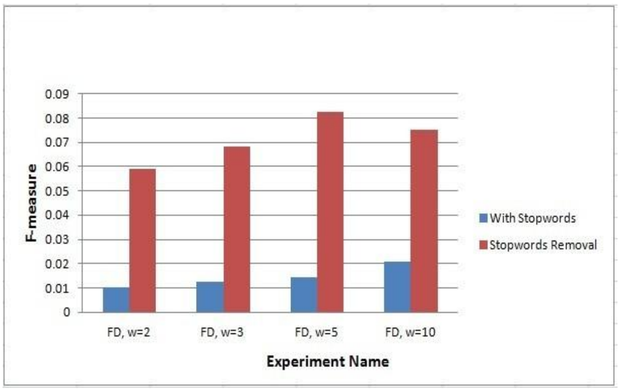


Fig. 8. F-measure using forward graph and different window sizes on training dataset with and without stopwords.

C. *Experiment 3: Effect of POS Tagging*

The third experiment was conducted to assess the effect of using POS tagging on the training dataset after the removal of stopwords.

This experiment involved removing the stopwords, tokenizing the words in the documents by utilizing the POS tagging process and then applying the proposed system using forward graph with window sizes 2, 3, 5 and 10. Table VIII shows the results.

Table VIII: TextRank Using Forward Graph (FD) various Window Sizes (w) on the Training Dataset after Removing Stopwords and Applying POS Tagging.

Experi-ment Name	Rele-vants	Ret-rieved	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measure
FD, w=2	401	982	107	0.2785	0.1093	0.1535
FD, w=3	401	987	104	0.2710	0.1055	0.1478
FD, w=5	401	986	101	0.2662	0.1028	0.1445
FD, w=10	401	989	103	0.2687	0.1048	0.1471

The effectiveness of POS tagging is shown in Fig. 9, which provides a comparison of the F-measure values for different window sizes of 2, 3, 5 and 10 using the forward graph on the training dataset after stopword removal with and without POS tagging.

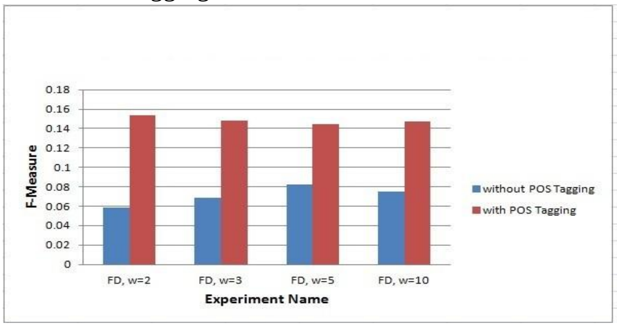


Fig. 9. F-measure using Forward graph with different Window Sizes on training dataset after removing Stopwords with POS tagging with and without POS tagging.

In order to assess the performance of the proposed system on the test dataset, the above features were applied to the test dataset at different window sizes. Table IX shows an example of the results when the features were applied to the test dataset with a window size of 2, which was found to be the optimum window size for this part of the experiments.

Table IX: TextRank Using Forward Graph (FD) on the test dataset after Removing Stopwords and Applying POS Tagging at a Window Size of 2.

Experi-ment Name	Rele-vants	Ret-riev ed	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measu re
FD, w=2	195	493	49	0.2855	0.1014	0.147 3

D. *Experiment 5: Effect of Using Weights for the Vertices in the Graph*

The fifth and final experiment involved adding different types of weight to the page rank score for each word based on several factors. Numerous runs of the experiment were made to find suitable weights that would improve the results of the proposed keyword extraction system. Three types of weights were used: Token frequency, token position (abstract or title) and a combination thereof.

1) *Token Frequency as a Weight*

In order to check the effect of token frequency on the training and test datasets, each page rank score for each word was multiplied by its frequency in a document. In Arabic texts words that are more frequent must be given more weight so that they are selected as keywords. Table X shows the results of applying the token frequency weight to the training dataset and Table XI shows the results for the test dataset. It can be seen that the F-measure score significantly increases from 0.1535 to 0.1683 for the training dataset and from 0.1473 to 0.1560 for the test dataset.

Table X: TextRank Using Term Frequency in the Forward Graph (FD) and Window Size of 2 on the Training Dataset after Removing Stopwords and Applying POS tagging

Experi-ment Name	Rele-vants	Ret-rieve d	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measur e
FD, w=2	401	982	117	0.3060	0.1203	0.1683

Table XI: TextRank Using Term Frequency in the Forward Graph (FD) and a Window Size of 2 on the Test Dataset after Removing Stopwords and Applying POS Tagging.

Experi-ment Name	Rele-vants	Ret-rieved	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measure
FD, w=2	195	493	52	0.3032	0.107	0.156

2) *Term Position as a Weight*

The automatic keywords and keyphrases generated by the proposed system and those that existed in the titles given more weights in order to determine whether this approach would have an effect on the results. Thus, every generated keyword or keyphrase in the title was multiplied by a number to give it more weight. Specifically, the page rank scores of the keywords and keyphrases in the titles were multiplied by weights of 1, 2, 3, 5, 10, 15, 20 and 30. Table XII shows the result of using these term positions as weights in the proposed system together with forward graph and a window size of 2 and applying the proposed system to the training dataset.

Table XII: TextRank Using Term Position Weights (WT) in the Forward Graph (FD) with a Window Size of 2 on the Training Dataset after Removing Stopwords and Applying POS Tagging.

Experi-ment Name	Rele-vants	Ret-rieved	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measu re
FD, WT=1	401	982	107	0.27 85	0.109 3	0.153 5
FD, WT=2	401	983	112	0.29 12	.1139	0.160 1
FD, WT=3	401	983	113	0.29 45	0.114 8	0.161 5

FD, WT=4	401	984	112	0.2912	0.1137	0.1598
FD, WT=5	401	984	112	0.2912	0.1137	0.1598
FD, WT=10	401	984	112	0.2912	0.1137	0.1598
FD, WT=15	401	984	112	0.2912	0.1137	0.1598
FD, WT=20	401	984	112	0.2912	0.1137	0.1598
FD, WT=30	401	984	112	0.2912	0.1137	0.1598

As can be observed from Table XII, increasing the term weight number resulted in an increase in the value of the F-measure. The weight number was increased until values became fixed and unchanged. The highest f-measure value was obtained by a weight number of 3. Table XIII shows the result of using the term position as a weight in the proposed system when it was applied to the test dataset.

It can be seen that very similar results were obtained by using several weights for the tokens in the titles. However, the highest F-measure value was still that obtained by the weight number of 3. So only this weight needs to be used to multiply the tokens that existed in the titles of the data sample.

Table XIII: TextRank Using Term Position Weights (WT) in the Forward Graph (FD) with a Window Size of 2 on the Test Dataset after Removing Stopwords and Applying POS Tagging.

Experi-ment Name	Rele-vants	Ret-rieved	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measure
FD, WT=1	195	493	49	0.2855	0.1014	0.1473
FD, WT=2	195	493	53	0.3078	0.1108	0.1604
FD, WT=3	195	493	53	0.3088	0.1112	0.1610
FD, WT=4	195	494	53	0.3088	0.1107	0.1605
FD, WT=5	195	494	53	0.3088	0.1107	0.1605
FD, WT=10	195	494	53	0.3088	0.1107	0.1605
FD, WT=15	195	494	53	0.3088	0.1107	0.1605
FD, WT=20	195	494	53	0.3088	0.1107	0.1605
FD, WT=30	195	494	53	0.3088	0.1107	0.1605

3) Term Frequency and Term Position as a Combined Weight

In a final experimental variation, the term frequency weight was combined with the term position weight, results to investigate whether there would be a further improvement in the value of the F-measure. Table XIV shows the result of using the combined term frequency and term position weights in the forward graph with a window size of 2 on the training dataset, after stopword removal and with POS. The weight number of 3 was multiplied with the tokens that existed in the titles.

Table XIV: TextRank Using Combination of Term Frequency and Term Position Weights (WT) in the Forward Graph (FD) with a Window Size of 2 on the Training Dataset after Stopword Removal and Application of POS Tagging.

Experi-ment Name	Rele-vants	Ret-rieved	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measure
FD, WT=3	401	684	111	0.2948	0.1936	0.2102

Table XV shows how using the above combination of weights in the proposed system also improved the results for test dataset using the same parameters.

Fig. 10 and 11 show the results of using the forward graph, stopword removal, POS tagging and a window size of 2 with the above combination of weights on the training dataset and test dataset, respectively.

Table XV: TextRank Using Combination of Term Frequency and Term Position Weights (WT) in the Forward Graph (FD) with a Window Size of 2 on the Test Dataset after Stopword Removal and Application of POS Tagging.

Experi-ment Name	Rele-vants	Ret-rieved	Ret-rieved Rele-vants	Recall	Pre-cesion	F-Measure
FD, WT=3	195	283	43	0.2465	0.1828	0.1982

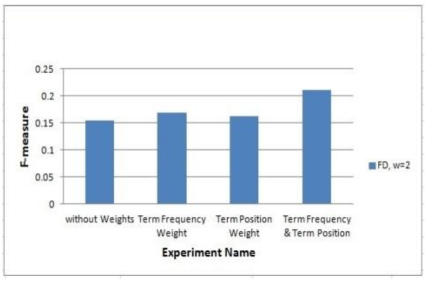


Fig. 10. F-measure using Forward graph and a window size of 2 on the training dataset after removing stopwords and applying POS tagging.

Fig. 11 shows the results of using the forward graph, stopword removal, POS tagging and a window size of 2 with the above weights on the test dataset.

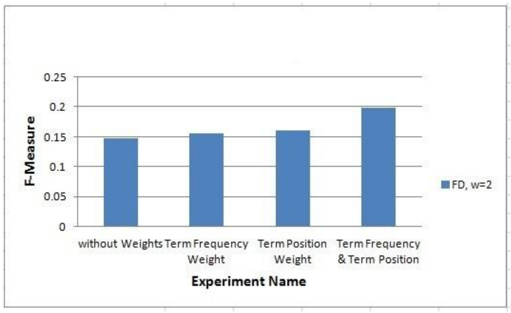


Fig. 11. F-measure for Different Experiments using Forward graph and a window size of 2 on the test dataset after removing stopwords and applying POS tagging.

V. CONCLUSION

The results of the experiments conducted for this study show that it is possible to build a keyword extraction system using the page rank algorithm and to apply it successfully to Arabic texts, despite the difficulties of the Arabic language which is morphologically rich and highly ambiguous due to its complex morpho-syntactic agreement rules and the presence of a lot of irregular word forms. The results of several experiments on the training dataset revealed the most suitable the suitable techniques and tools to use to obtain the best possible results when applying the proposed keyword extraction system to the test dataset. These techniques and tools are as follows:

- Using the forward graph
- Removing stopwords from documents as a preprocessing step.
- Applying a linguistic feature, namely POS tags, on the words in documents as a preprocessing step.
- Using the window size of 2.
- Multiplying the page rank scores of the graph vertices by their frequencies and multiplying the page rank scores of the terms in the titles by 3.

The results showed that removing all meaningless stopwords, and nominating general nouns that exist either in the abstracts or titles improves the results significantly. In addition, the results proved that the word position is an important factor in a keyword extraction system for Arabic text as assigning more weight to the words that exist in titles improve the results significantly. Furthermore, the results indicated that the word frequency is also an important factor as multiplying each word's page rank score by its frequency also led to a significant improvement in the results.

REFERENCES

1. Al-Muhtaseb, H., and Mellish, C. 1998. Some differences between Arabic and English: A step towards an Arabic upper model. In Proceedings of the 6th international conference on multilingual computing pages 118-45. Cambridge, UK.
2. Brin, S., and Page, L. 1998. The anatomy of a large-scale hyper-textual web search engine. Computer Networks and ISDN Systems. Pages 107-17.
3. Kouninef, B., AL-Johar, B. 2011. Extracting entities and relationships from Arabic text for information system. Journal of Emerging Trends in Computing and Information Sciences. Pages 641-5.
4. Habash, N., Soudi, A., and Buckwalter, T. 2007. Arabic computational morphology, Knowledge-based and empirical methods. In (chap. 2). Springer Netherlands.
5. Hulth, A. 2003 Textrank: Bringing order into texts. In Proceedings of the 2003 conference on empirical methods in natural language processing. Pages 216-23. Japan.
6. Lawrence, P., Sergey, B., Motwani, R., Winograd, T. 1998. The PageRank citation ranking: Bringing order to the web (Tech. Rep.). Stanford University.
7. Lofgren, P., Banerjee, S., and Goel, A. 2015. Personalized PageRank Estimation and Search: A Bidirectional Approach. ArXiv:1507.05999v3 [cs.DS]
8. Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into texts. In Proceedings of empirical methods on natural language processing conference.
9. Mihalcea, R., Liu, H., and Lieberman, H. 2006. NLP (Natural Language Processing). In 7th international conference, cicling. Pages 319-30.
10. Namly, D., Bouzoubaa, K., Tahir, Y., and Khamar, H. 2015. Development of Arabic particles lexicon using the LMF framework. Colloque pour les Etudiants Chercheurs en Traitement Automatique du Language Naturel ET SES applications (CEC-TAL 2015) Sousse - Tunisi e, le 23-5.
11. Rose, S., Engel, D., Cramer, N., and Cowley, W. 2010. Automatic keyword extraction from individual documents. In Text mining. Applications and theory. Pages 1-20. John Wiley and Sons, Ltd.
12. Sarkar, K. 2013. A Hybrid Approach to Extract Keyphrases from Medical Documents.
13. Sarkar, K., Nasipuri, M., and Ghose, S. 2010. A new approach to keyphrase extraction using neural networks. IJCSI International Journal of Computer Science.
14. Sarmento, R., Cordeiro, M., Brazdil, P., and Gama, J. 2018 Incremental TextRank - Automatic Keyword Extraction for Text Streams. Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018) - Volume 1, pages 363-70.

AUTHORS PROFILE

Meran M. Al Hadidi is currently working as an instructor at Al-Balqa Applied University in Princess Alia University College, Shmeisani, Amman, Jordan. She received her B.Sc. Degree in computer science from University of Jordan. She received her High Diploma Degree in computer science from Al-Balqa Applied University. And received her M.Sc. degree in computer science from Al-Balqa Applied University. Her researches include Natural Language processing and Digital Image Processing. Having experience in Java, Perl, Python and MATLAB programming languages. She can be reached at Email: meran_hadidi@bau.edu.jo

Dr. Muath Alzghool is currently working as a post-doctoral visitor at York University, Canada in Electrical Engineering and Computer Science Department. He received his B.Sc., and M.Sc. Degrees in computer science from Yarmouk University, Jordan. He received his Ph.D. degree in computer science from University of Ottawa, Canada, in 2009. He worked at Al-Balqa Applied University/Jordan as an assistant professor. His research interests include Artificial Intelligence, Natural Language Processing, Information Retrieval, Speech Recognition, and Data Mining. He has Experience in Java and Perl programming languages. He can be reached at Email: alzghool@yorku.ca

Dr. Hasan Muaidi is currently working as an associate professor at Al-Balqa Applied University, in Faculty of Huson College, Irbid, Jordan. He received his B.Sc., and M.Sc. degrees in computer science from Yarmouk University, Jordan, in 1987 and 2002, respectively, and his Ph.D. degree in Artificial Intelligence and Software Engineering from DeMontfort University, Leicester, U.K, in 2008. His research focuses on Artificial Intelligence, Artificial Neural Networks, Arabic Natural Language Processing, Information Retrieval, Expert Systems, Machine Learning and Software Engineering. He can be reached at Email: alserhan@bau.edu.jo