Automatic Keyphrase Extractor from Arabic Documents

Hassan M. Najadat
Department of Computer information Systems
Jordan University of Science and Technology
Irbid, Jordan

Mohammed N. Al-Kabi
Computer Science Department
Zarqa University
Zarqa, Jordan

Ismail I. Hmeidi
Department of Computer information Systems
Jordan University of Science and Technology
Irbid, Jordan

Maysa Mahmoud Bany Issa
Computer Science Department
Jordan University of Science and Technology
Irbid, Jordan

***Abstract*—The keyphrase is a sentence or a part of a sentence that contains a sequence of words that expresses the meaning and the purpose of any given paragraph. Keyphrase extraction is the task of identifying the possible keyphrases from a given document. Many applications including text summarization, indexing, and characterization use keyphrase extraction. Also, it is an essential task to improve the performance of any information retrieval system. The internet contains a massive amount of documents that may have been manually assigned keyphrases or not. The Arabic language is an important language in the world. Nowadays the number of online Arabic documents is growing rapidly; and most of them have no manually assigned keyphrases, so the user will scan the whole retrieved web documents. To avoid scanning the entire retrieved document, we need keyphrases assigned to each web document manually or automatically. This paper addresses the problem of identifying keyphrases in Arabic documents automatically. In this work, we provide a novel algorithm that identified keyphrases from Arabic text. The new algorithm, Automatic Keyphrases Extraction from Arabic (AKEA), extracts keyphrases from Arabic documents automatically. In order to test the algorithm, we collected a dataset containing 100 documents from Arabic wiki; also, we downloaded another 56 agricultural documents from Food and Agricultural Organization of the United Nations (F.A.O.). The evaluation results show that the system achieves 83% precision value in identifying 2-word and 3-word keyphrases from agricultural domains.**

***Keywords*—Arabic Keyphrase Extraction; Unsupervised Arabic Keyphrase Extraction; Information Retrieval**

I. INTRODUCTION

The world witnessed during the last two decades an exponential growth in the size of the Internet, which represents the largest heterogeneous reservoir of information. Web documents contain information stored in this global system of interconnected computer networks which is called the Internet. Information stored in the Internet varies in their type, where we can find text, audio, video, images, and other formats.

The Arabic language is one of the six official languages adopted by the United Nations since it ranked the fifth largest natural language among the top 100 used natural languages worldwide. But Arab Internet users ranked 7th worldwide following the users of the following languages, English, Chinese, Spanish, Japanese, Portuguese and German. Arabs constitute 5% of the world population while their Arabic content constitutes only 1% of the Internet content. Although Arab contribution to the Web is one fifth of their population estimates, but on the Internet, there is a large number of Arabic textual documents stored in this giant reservoir of information. Keyphrase extraction is an essential process in information retrieval, document summarization, and clustering. We can extract keyphrases either manually or automatically. Some of the Web textual articles have manually extracted keyphrases. Also, the effectiveness of manual keyphrase extraction is higher than its counterpart automatic keyphrase extraction, but it is costly and slow about automatic keyphrase extraction.

Some studies are conducted to explore the automatic extraction of Arabic keyphrases. This study presents a new unsupervised algorithm to extract Arabic keyphrases from textual documents, where attributes such as Term Frequency-Inverse Document Frequency (TF×IDF), Phrase position, title threshold, terms frequency, phrase frequency, and phrase distribution are used by this novel algorithm to identify keyphrases.

This study is organized as follow: Section 2 presents an overview of the related work to Keyphrase extraction while Section 3 presents the methodology followed to accomplish this study Section 4 presents the results of the tests conducted on our new algorithm while Section 5 presents conclusion remarks and future work.

II. RELATED WORKS

First, this section presents a review of few numbers of related studies to our new algorithm. Witten, Paynter, Frank, Gutwin and Nevill-Manning study presents an automatic algorithm called Kea to extract keyphrases from textual documents. Kea uses lexical methods to identify candidate keyphrases, where a score is computed for each candidate keyphrase. Also, Kea adopts machine learning techniques to

identify the good candidate keyphrases. Tests were conducted on their algorithm using a large dataset yield a good performance [6].

An interactive tool called PhraseRate to help human classifiers in the Infomine Project is presented by J.B. Keith Humphreys. This tool requires no training and uses Webpage structure to extract keyphrase from those Web pages, where tests on this tool prove its effectiveness [4].

A statistical language model is used by Takashi Tomokiyo and Matthew Hurst to extract keyphrases, where phraseness and informativeness unified into a single score to rank the automatically extracted keyphrases [5]. Turney et al. 2003 [13] exhibit an approach to extract keyphrases, where each document is decomposed into a number of phrases. Each of these phrases is considered as a candidate keyphrase. A supervised learning algorithm is used to identify keyphrases. Another study conducted by Medelyan et al. 2009 [9] shows that providing high- quality features to machine learning algorithm will lead to successfully extracting keyphrases.

Min Song et al. 2003 [8] demonstrate KPSpotter which provides flexible and web-enabled keyphrase extraction by combining the information-Gain data mining measure with multiple NLP methods. This algorithm processes multiple input text formats such as HTML or XML. TF×IDF and distance are measured from first occurrence. Then the attributes are discretized into ranges to calculate the probability of each candidate phrase to be a keyphrase. According to these values, the candidate phrases are ranked to select the most descriptive candidate phrase to be a keypharse. The algorithm was tested on a set of abstracts of some technical reports. Although the experiments showed that both KPSpotter and KEA perform poorly in terms of an average number of matches because of document length, both produce phrases with equal quality.

Quanzhi Li et al 2005 [11] provides a domain specific keyphrase extraction program called Keyphrase Identification Program (KIP). This program extracts a list of candidate noun phrases based on logic. Then, the algorithm sets a score for each term in each candidate phrase. A human-developed glossary database is used to store domain specific keywords and keyphrases and their initial weights. This database contains two tables, one for keyphrase and the other one for keyterm. Each table stores the keyphrase/keyterm and its weight. At first, the keyphrases and terms with their initial weights are defined manually. Then, the learning process takes its role which can be automatic or user-involved. By involving the user in the learning process, the quality of keyphrases can be controlled by the user of the program, he/she can add, remove and highlight any keyphrase he/she wants and the program will respond to that personalization feature.

Samahaa R. El-Beltagy and Ahmed Rafea 2009 [12] propose efficient extraction system for English language called KP-Miner, which uses the simplest version of Poter's stemmer, also they provide adaptation to the system to be able to work with Arabic documents. Although the system does not need training to achieve the extraction task, it was proved by experiments, that the system does good job that is comparable with KEA algorithm.

Also the study conducted by Jiang et al. 2009 [16] emphasize on the importance of using learning by rank techniques to extract keyphrases. Those researchers proposed casting the keyphrase extraction problem as ranking and learning, rather than casting it as a classification (keyphrases and non-keyphrases) using decision tree and Naive Bayes classifiers. Their experiments show that SVM significantly outperforms the others, where learning is exploited. Furthermore, Liu et al. 2010 [19] propose using a Topical PageRank (TPR) on word graph to determine the word importance with respected to different topics. Afterword the distribution of topics within each document is determined, and then the ranking scores of each extracted word are computed. Finally, the top ranked words are considered keyphrases by this method.

Liu et al. 2009 [18] propose unsupervised clustering based method for keyphrase extraction. Using clustering method on a document leads to a creation of different clusters, where the clustering starts with exemplar terms representing the centroid of each newly created cluster, and then all semantically related words and phrases are grouped into a single cluster. They claim that their newly proposed method outperform the sate-of-the-art graph-based ranking methods (TextRank) by 9.5% in F1-measure.

A study is conducted by Wan et al. 2010 [15] proposes the use of a few number of nearest neighbor documents to each document to enhance the process of document summarization and keyphrase extraction. To apply this cornerstone idea a graph-based ranking algorithm is used, where this algorithm uses local information extracted from the document under consideration, and global information extracted from neighbor documents. The tests show clearly the effectiveness and robustness of their proposed method.

According to Alexa, social networking sites like Facebook, Youtube, Twitter, LinkedIn are globally top ranked [1]. A huge number of messages, comments, and views are exchanged within social networking sites. Analyzing this huge number of messages and comments manually is tedious, slow, expensive, and impractical. A study by Zhao et al. 2011 [17] proposes a context-sensitive topical PageRank (cTPR) method to rank different keywords and extract topical keyphrases from Tweeter short messages (Tweets) [14]. This novel method uses a probabilistic scoring function to determine the relevance and interestingness of each keyphrase. Tests show the effectiveness of this method to extract topical keyphrases. Zhao et al. [17] represents an improvement to Liu et al. 2010 [19] study in which they propose using a Topical PageRank (TPR).

El-Beltagy et al. 2009 [12] exhibit in their study a new system to extract Arabic/English keyphrases from textual documents. Their system is called KP-Miner, which needs no training, and characterizes by an equivalent accuracy and sometimes superior to the accuracy of supervised machine learning systems [10, 14] used to extract keyphrases.

On the other hand El-shishtawy et al. 2009 [3] study used supervised learning techniques to extract Arabic keyphrases from Arabic documents. They used a method that does not rely on statistical information such as Term Frequency (TF)

and term distances, but relies on linguistic knowledge, which includes syntactic rules based on part of speech (POS) tags. This helps to extract candidate keyphrases. Linear discriminant analysis (LDA) method is used to find a linear combination of linguistic features characterizing keyphrases, where ANOVA (analysis of variance) is used to evaluate each of the selected features. Tests show the effectiveness of this method to extract Arabic Keyphrases.

Al-Kabi et al. 2012 [2] study is based mainly on the Term Frequency (TF) to identify top frequent terms to build a co-occurrence matrix showing the occurrence of each frequent term. If the term is in the biasness degree, then the term is important, and could be considered as a candidate to be a keyword. Words with high $\chi 2$ could be considered a probable keyword, and $\chi 2$ proves it is better to identify keywords than a novel method based on term frequency - inverted term frequency (TF-ITF).

III. METHODOLOGY

B This part of the study presents the necessary steps followed to extract Arabic keyphrases extracted from the collected Arabic documents. In this study, around 200 Arabic Web documents collected from Wikipedia website (http://www.wikipedia.org/) and the Website of UN Food and Agricultural Organization (FAO) are used. Fig.1 presents the algorithm of our proposed System (AKEA) which used in this study to extract Arabic Keyphrases.

Consider the following notes related to algorithms shown in Fig.1: The Phrase (Ph) will be nominated as a candidate phrase if its frequency (PF) exceeds 2, since the Keyphrase in Arabic language must exist at least twice within a single paragraph.

After identifying each Arabic Keyphrase in the collection, the following attributes of each candidate Keyphrase are extracted: phrase frequency (PF), summation of phrase terms frequencies (Tf), PF×IDF (Phrase Frequency–Inverse Document Frequency), Phrase Position (Ph_Pos), Title Threshold (T_Thresh) and phrase distribution (Ph_Dist).

Eq. (1) represents PFScore which uses all the attributes mentioned in the previous paragraph. The equation is deduced empirically during conducting a series of tests to extract Arabic Keyphrases.

$$PF_{Score} = \left(\frac{1}{Ph\_Pos + 1}\right) + T\_Thresh + \sum_{i=1}^{Ph\_Len} TF + (Ph\_Dist) + \log_2 PhF + (PhF \times IDF) \quad (1)$$

Eq. (1) is a combination of adding a number of terms on the right- hand side of Eq. (1). The first term is (1/(Ph_Pos+1)), which represents the reciprocal of Phrase Position, Ph_Pos, plus one to avoid division by zero. This term yields the highest score to phrase at the beginning of each paragraph. This term is based on the idea that Arabic keyphrases lie in most cases at the beginning of each paragraph.

The second term on the RHS within Eq. (1) is T_Thresh. This term yields highest scores to those keyphrases which contain all the terms in the document title.

```
Algorithm: AKEA.
Input: Arabic Textual Document.
Output: List of the Extracted Arabic Keyphrases.
BEGIN
  WHILE Not EOF
   Remove Arabic Stop Words
   Stem Arabic Text
   Compute Term Frequency (TF) of each Arabic
   Identify each Paragraph P in the document
   WHILE NOT END of (P)
     Identify each Phrase Ph in the document
     Compute Phrase Frequency (PF)
     IF (PF) > 2
       Extract Phrase (Ph) attributes
       Compute Phrase score (Pscore)
       Save P, Ph, PF, and Pscore into (Phrases-List)
     END IF
   END WHILE
   WHILE NOT END of Phrases-List
    IF PF > 1
     Choose the highest frequency phrase
    END IF
    IF Ph is a Substring from any phrase in Phrases-list
     Remove Ph from the Phrases-List
    END IF
  END WHILE
  ENDWHILE
  Rank candidate phrases Ph in Phrases-List in descending order
  according to their PFScore
END
```

Fig. 1. Proposed AKEA Algorithm

The third term in Eq. (1) is the summation of term frequencies of the words which the phrase under consideration is consisting of summation keyphrases mostly contains high-frequency words. The expressive words are repeated over all the text. In this term, Ph_Len represents phrase length.

The fourth term in Eq. (1) is Phrase Distribution, Ph_Dist, which gives the probability of the phrase to be appearing in the i^{th} paragraph. So the phrase that has the highest distribution will be the most descriptive one to explain the idea of the paragraph. The frequency of the phrase helps in selecting the candidate phrases and keyphrases. For the keyphrases, they should repeat more than twice in the paragraph. All of the attributes are necessary and each one gives valuable information about the phrase, so that the output of the experiment will be more accurate.

The fifth term in Eq. (1) is $\log_2 PhF$, where PhF represents a ratio computed according to Eq. (2):

$$PhF = \frac{Doc\_PhF}{Doc\_Total\_Ph} \quad (2)$$

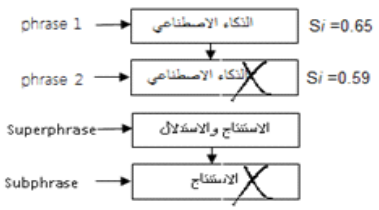

Fig. 2. Example of removing some phrases from candidate phrases list

Where Doc_PhF is a specific phrase frequency in a document, and Doc_Total_Ph is the total number of phrases in that document. The sixth term in Eq. (1) is PhF × IDF, which is the product of the previous ratio (PhF) used in the fifth term by inverse document frequency (IDF). After extracting the phrases of each paragraph and compute the score of each phrase. Some phrases may be repeated more than once if the system extracts the same phrase from different paragraphs. If the phrase exists in the phrases list more than once, the system will choose the highest score phrase and drop the duplicates. Also, it will drop the sub-phrases of some super-phrases to get the final candidate phrases list. Fig. 2 presents two examples that explain how to drop the duplicate phrases and sub-phrases.

IV. Experimental Results

Most of Keyphrase extraction systems must be trained before it can be applied to new documents. In our work, the system will not depend on training because of a large variety of subjects and we do not use domain-specific documents. In this section, we provide the results of our algorithm to extract keyphrases from Arabic documents. We will provide different combinations of the attributes that we used to define the score of the phrases and compare their performance. The performance of individual attributes differs completely from the performance of different combinations of attributes of AKEA system. This is what will be shown in the remaining of this section.

A. Different Attributes Combinations

Different combinations of the attributes are provided in this section. The individual attributes which were used in Eq. (1) are: phrase frequency, terms frequency, title threshold, TF×IDF, phrase position and phrase distribution. Using the attributes individually is not beneficial. Single attribute of a phrase does not give any indication about the importance of the phrase in the document. So we try many different combinations of these attributes and compare their results. For each combination of attributes, we compute the mean value of the results of the 100 documents of the dataset.

B. Single Attribute Performance

Table 1 shows the performance of different attributes individually in identifying different number of phrases. In this table, the column of number of correct keyphrases displays the fraction of automatic keyphrases over the manual keyphrases, while the column of number of phrases displays how many phrases that chosen from the top ranked phrases. Fig. 3 shows the random behavior for the system which tends to decrease in the average precision value. So we suggest new combinations of attributes that give better results. Now we give some examples of the different combinations and their results.

1) Two Attributes Combinations

In this section, we give the performance of different combinations of the five attributes: term frequency, title threshold, TF×IDF, position and distribution with phrase frequency as an example of combining two attributes at a time. Table 2 shows the details of combining phrase frequency with other attributes, one at a time. Fig. 4 shows the relationship between the number of phrases and the precision value for each combination mentioned in Table 2.

TABLE I. Performance of Different Attributes Individually (Experiment 1)

Attribute	Number of keyphrases	Number of correct phrases
Phrase Frequency	1	0.33
	5	0.41
	10	0.43
Terms Frequency	1	0.27
	5	0.35
	10	0.47
Title threshold	1	0.25
	5	0.34
	10	0.39
Pf×idf	1	0.37
	5	0.4
	10	0.41
Position	1	0.24
	5	0.29
	10	0.38
Distribution	1	0.35
	5	0.36
	10	0.44

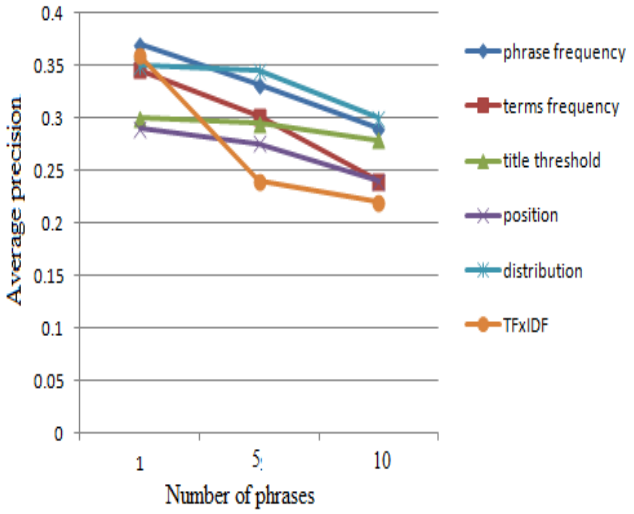

Fig. 3. Comparison of the individual performance of different attributes

TABLE II. Performance of Combining Two Attributes at a Time (Experiment 2)

Combination	Number of keyphrases	Number of correct phrases
Phrase frequency + term frequency	1	0.33
	5	0.41
	10	0.43
Phrase frequency +Title threshold	1	0.25
	5	0.34
	10	0.39
Phrase Frequency+PF×IDF	1	0.37
	5	0.4
	10	0.41
Phrase Frequency+Position	1	0.24
	5	0.29
	10	0.38
Phrase Frequency+Distribution	1	0.35
	5	0.36
	10	0.44

The information that presented by Fig. 4 confirms that we have to explore other combinations. The highest value of average precision appears when we take the top ten ranked phrases by using phrase frequency and distribution, but we may get a higher value of precision if we try other combination. If we try to combine two attributes at a time we need 15 experiments which are difficult to be explained.

2) Three Attributes Combinations

The example that we choose randomly to use here is to combine phrase frequency and phrase position with one attribute at a time from the following four attributes: term frequency, title threshold, TF×IDF and the phrase distribution. Table 3 shows the number of correct phrases for each combination. Keep in mind that the number of correct phrases is equal to the number of correct keyphrases that identified automatically divided by the number of manually identified keyphrases.

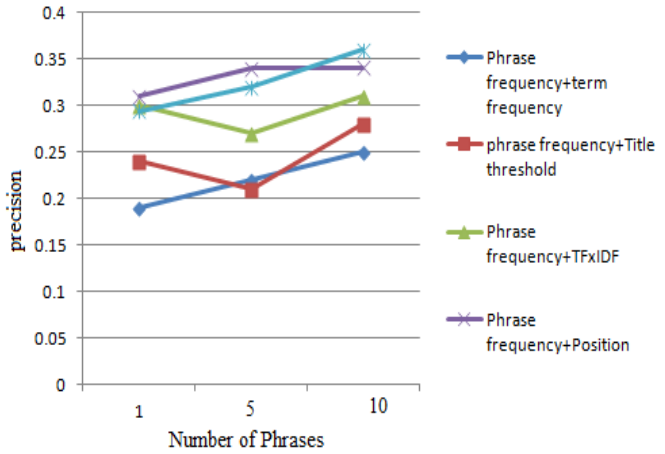

Fig. 4. Phrase frequency combinations performance

TABLE III. PERFORMANCE OF 3-ATTRIBUTE COMBINATIONS (EXPERIMENT 3)

Combination	Number of keyphrases	Number of correct phrases
Phrase frequency + position + term frequency	1	0.35
	5	0.4
	10	0.42
Phrase frequency + position + Title threshold	1	0.29
	5	0.35
	10	0.39
Phrase frequency + position + PF×IDF	1	0.39
	5	0.42
	10	0.42
Phrase frequency + Position + distribution	1	0.4
	5	0.43
	10	0.45

Fig. 5 shows a comparison between the precision values for each combination mentioned in Table 3. The experiments that we mentioned above shows a very convergent precision values except the combination phrase_frequency + position + distribution. This combination gives the highest precision value in increasing manner, but we still need a higher value for precision. For that reason, we try to find an equation that utilizes the advantages of all of the six attributes and combine them together, because all of the attributes are important. In this case no need to try different combinations.

3) The Best Combination

Each attribute has its own value that express information about the phrase. Phrase frequency gives the number of occurrences of the phrase in a given paragraph. It is common that the more important phrase will be redundant more than twice in the paragraph. Term frequency attribute represents the summation of phrase terms frequencies. Title threshold gives a value that expresses the relatedness between the phrase and the title of the document. PF×IDF is the combination between phrase frequency (PF) which is the number of occurrences of a specific phrase in a specific document, and inverse document frequency (IDF) which is the log of the ratio between a number of documents in the collection and number of the documents containing a specific phrase.

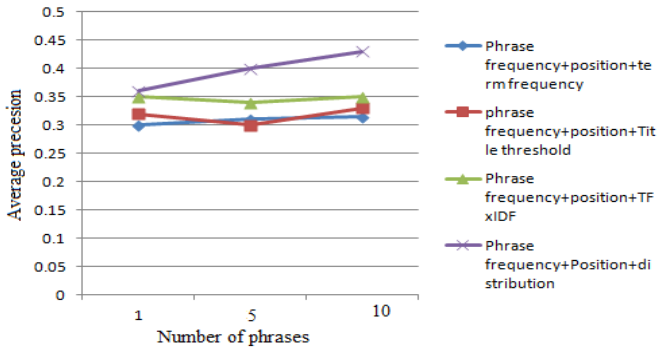

Fig. 5. 3-attribute combinations precision values

The value PF×IDF in our experiments is not very useful since we have non homogeneous document collection. The phrase position attribute is the number of words that precede the first appearance of the first word of the phrase in the paragraph. Lastly, the phrase distribution attribute is the possibility of the phrase appearing in the ith paragraph. We investigate the result of Eq. (1) and display them in Table 4 and Table 5. The value of phrase score PFscore represents the importance of the phrase in a specific paragraph. Fig. 6 presents the results of identifying 2-word keyphrases from stemmed and unstemmed text. Using Eq. (1) we get 0.7 average precision from stemmed text which is the best result of all experiments

TABLE IV. THE PERFORMANCE OF SI EQUATION FOR UNSTEMMED DOCUMENTS

Combination	Number of keyphrases	Number of correct phrases
Si	1	0.43
	5	0.47
	10	0.52

TABLE V. THE PERFORMANCE OF SI EQUATION GOR STEMMED DOCUMENTS

Combination	Number of keyphrases	Number of correct phrases
Si	1	0.54
	5	0.59
	10	0.67

Fig. 7 presents the average precision values the system achieved to identify 2-word and 3-word keyphrases from stemmed and unstemmed datasets.

It is clear that the number of correct phrases and precision values are raised obviously with the top 10 identified keyphrases. The AKEA system has achieved 70% accuracy using precision measure overall 100 test documents in identifying 2-word phrases. Also, it achieved 51% accuracy of precision measure in identifying 3-word keyphrases.

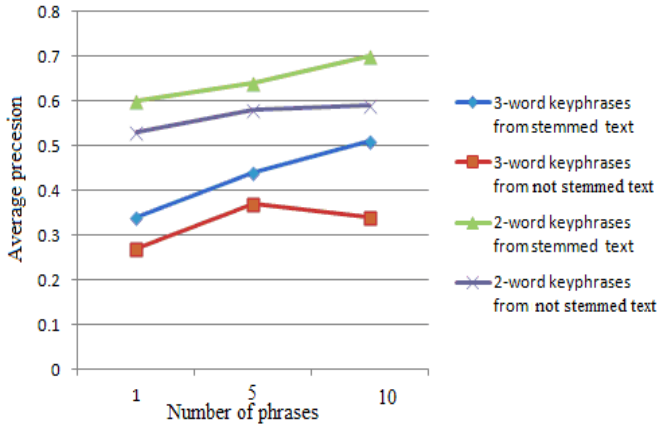

Fig. 6. Comparison between identifying 2-word and 3-word keyphrases from stemmed and not stemmed text

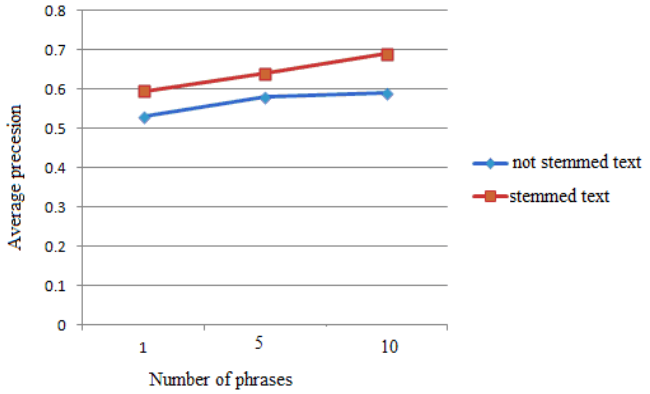

Fig. 7. Comparison between stemmed and unstemmed text output

The final results show that the AKEA system achieved 61% average accuracy of precision measure in identifying 2-word and 3-word keyphrases over all the 100 test documents.

The textual resources that had been used in our project were collected from Wikipedia website. The collection consists of 100 full-text documents and their abstracts that had been randomly downloaded from Arabic wiki. For each document, we run the system twice including using the stemmer [7] and without the stemmer in order to compare the behavior of the system in both cases. After getting the output for each document, we compare the results with the manually extracted phrases. The document collection that had been used to test the results of AKEA system was downloaded from www.ar.wikipedia.org. It contains 100 full-text documents with their abstracts from various domains. This document collection had been used to test KP-miner system [12]. A dataset of our documents and their manual keyphrases is available on www.claes.sci.eg/coe_wm/Data.htm. The average number of words per document in the dataset is in a range between 804 and 934 [12].

Majority of websites such as IEEE (Institute of Electrical and Electronics Engineers) that provides electronic documents provides only the abstract of the documents. AKEA system deals with the abstract like a paragraph, so it can identify keyphrases from any text regardless of the parts. Furthermore, the electronic documents provided by some websites from the types HTML and XML contain HTML/XML tags. These tags are removed by AKEA because they are non-Arabic letters and symbols provided that the input of our system must be a text file from utf-8 format. To investigate the behavior of the system when we provide it with an input that contains HTML/XML tags, a set of documents also downloaded from www.claes.sci.eg/coe_wm/Data.htm. We also test AKEA algorithm on another dataset contains 56 agricultural documents downloaded from FAO.

C. Evaluation Criteria

Using the author-assigned keyphrases as a gauge for assessing automatic-extracted keyphrases is logical suggestion because it eases the comparison between both keyphrases groups. Keep in mind author-assigned keyphrases are ranked by their importance, so it will help in evaluating the automatically extracted keyphrase quality. Table 6 shows examples to explain how to assess the keyphrase quality criteria. The column named system phrase contains the phrase identified by the system as a keyphrase, author phrase is the phrase that assigned manually as a keyphrase by the author of the document. The assessment column tells how to assess the system phrase, if the assessment is similar the system phrase is correct keyphrase, otherwise it is incorrect.

1) Precision and Recall

Precision and recall are the most famous measure to evaluate the information retrieval systems. When evaluating IR system, the precision is the fraction of retrieved document that are relevant, while recall is the fraction of all relevant documents retrieved. Table 7 explains all the possibilities of a given document in the dataset in an information retrieval system. The measures in Eq. (3) and Eq. (4) are used to evaluate the performance of information retrieval system. In keyphrase extraction system, any phrase might be keyphrase or non-keyphrase identified by the system. In addition, the document author might identify the phrase as keyphrase or non-keyphrase. So we have four possible cases of any phrase. Table 8 shows these possible cases.

According to Table 8, the definition of precision and recall will be as follows: Precision is the ability to retrieve top-ranked phrases that are most relevant. It is the proportion of extracted keyphrases that are correct. It can be calculated according to the following equation: P=A/(A+B), where A represents a number of keyphrases identified automatically and manually, and B represents a number of keyphrases identified automatically but not manually. Recall is the ability of the search to find all relevant phrases in the document. In keyphrase identification systems recall is defined as the proportion of correct keyphrases extracted.

TABLE VI. EXAMPLES OF ASSESSING THE SYSTEM IDENTIFIED PHRASES

System Phrase	Author Phrase	Assessment	Reason of assessment
حاسوب Computer	حساب Computing	Similar	Both phrases have the same stem (compute حسب).
حاسوب Computer	علم الحاسوب Computer Science	Different	The superphrase (علم الحاسوب) gives different meaning from the subphrase (الحاسوب).
علم الحاسوب Computer Science	علم- الحاسوب Computer-Science	Similar	The use of hyphen (–) and the slash (/) is allowed in the middle of the phrase.
علم الحاسوب Computer Science	علم، الحاسوب Computer, Science	Different	Using punctuation is not allowed in the middle of the phrase.
علم الحاسوب Computer Science	ع ح CS	Different	The abbreviation is different from the phrase.

TABLE VII. DOCUMENT CASES

	Relevant	Irrelevant
Retrieved	A	B
Not retrieved	C	D

$$Precision = A / (A+B) \quad (3)$$

$$Recall = A / (A+C) \quad (4)$$

TABLE VIII. PHRASE CASES

	Identified as keyphrases by the author	Identified as Non-keyphrases by the author
Identified as keyphrases by system	A	B
Identified as Non-keyphrases by system	C	D

The following equation calculates the recall value: R=A/(A+C), where A represents a number of keyphrases identified automatically and manually, and C represents a number of keyphrases identified manually but not automatically.

2) Results

In this section, we provide the results of our algorithm to extract keyphrases from Arabic documents according to our experiments that were explained in the previous subsections. Using the attributes individually is not beneficial. A Single attribute of a phrase does not give any indication about the importance of the phrase in the document. So we use many different combinations of these attributes and compare their results. For each combination of attributes, we compute the mean value of the results of the 100 documents of the dataset. The conducted tests on AKEA using the two types of Arabic documents (stemmed and not-stemmed), that it is better to stem Arabic text before using Arabic Keyphrase extractor as shown in Fig. 8. Fig. 9 presents a comparison between AKEA algorithm and KP-miner in extracting 2-word key-phrases using the same dataset.

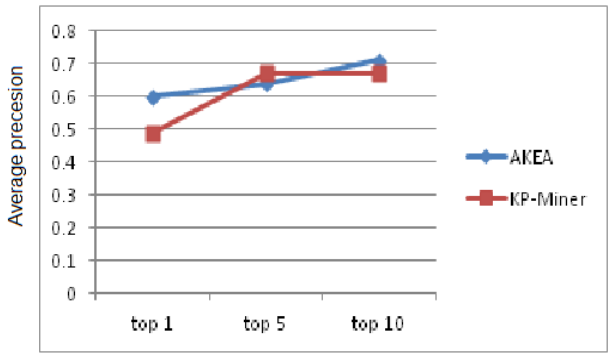

Fig. 8. Identifying 2-word key-phrases in AKEA and KP-Miner

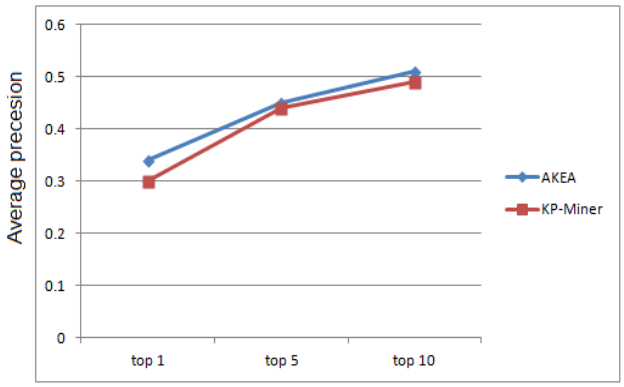

Fig. 9. Identifying 3-word keyphrases with AKEA and KP-Miner

To test the effectiveness of the AKEA algorithm to extract key-phrases from a domain-specific dataset, a collection of 56 various agricultural documents were collected from Food and Agriculture Organization of the United Nations (FAO) Website (http://www.fao.org). AKEA yields 83% precision to extract the top 10 key-phrases from this agricultural collection.

V. CONCLUSION

This study presents a novel supervised Arabic key-phrase detection algorithm using a limited dataset of around 200 Arabic Web documents collected from Arabic Wikipedia and Food and Agricultural Organization of the United Nations (FAO). This algorithm yields satisfactory accuracy results.

Future work includes the use of a larger dataset to test an enhanced version of our proposed algorithm, where new attributes will be adopted to improve the effectiveness of this algorithm.

REFERENCES

[1] Alexa Top 500 Global Sites, Available at: http://www.alexa.com/topsites (Accessed September 9, 2015).

[2] M. Al-Kabi, H. Al-Belaili, B. Abul-Huda, and A. Wahbeh, "Keyword extraction based on word co-occurrence statistical information for arabic text", Abhath Al-Yarmouk:Basic Science & Engineering., 22 (1), pp. 75- 95, 2013.

[3] T. El-Shishtawy and A. Al-Sammak," Arabic keyphrase extraction using linguistic knowledge and machine learning techniques", Proceedings of the Second International Conference on Arabic Language Resources and Tools, 2009.

[4] K. Humphreys, "PhraseRate: An HTML keyphrase extractor", Technical report, University of California, 2002.

[5] H. T. Tomokiyo and M. Hurst, “A language model approach to keyphrase extraction”, Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment, vol. 18, pp. 33-40, 2002.

[6] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G Nevill-Manning, “KEA: practical automatic keyphrase extraction”, Proceeding of the fourth ACM conference on Digital libraries, pp. 254-255, 1999.

[7] Java version of Shereen Khoja Arabic stemmer, Available at: http://zeus.cs.pacificu.edu/shereen/ArabicStemmerCode.zip (Accessed September 9, 2015).

[8] M. Song, I. Song, and X. Hu, “KPSpotter: a flexible information gainbased keyphrase extraction system”, Proceedings of the 5th ACM international workshop on web information and data management, pp. 50-53, 2003.

[9] O. Medelyan, E. Frank, and I. H. Witten, “Human-competitive tagging using automatic keyphrase extraction”, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: vol. 3. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1318-1327, 2009.

[10] P. D. Turney, “Learning algorithms for keyphrase extraction”, Information Retrieval, 2(3), pp. 303–336, 2000.

[11] L. Quanzhhi, W. Y. Brook, B. R. Stefan, and C. Xin, "Automatically finding significant topical terms from documents", AMCIS 2005 Proceedings, 2005.

[12] S. R. El-Beltagy and A. A. Rafea, “KP-Miner: A keyphrase extraction system for English and Arabic documents”, Inf. Syst., (34), pp. 132-144, 2009.

[13] P. D. Turney, “Coherent keyphrase extraction via web mining”, Proceedings of the 18th international joint conference on Artificial intelligence, pp. 434-439, 2003.

[14] Twitter, Available at: http://twitter.com/ (Accessed September 9, 2015).

[15] X. Wan and J. Xiao, “Exploiting neighborhood knowledge for single document summarization and keyphrase extraction”, ACM Trans. Inf. Syst, 28, 2, 2010.

[16] X. Jiang, Y. Hu, and H. Li, "A ranking approach to keyphrase extraction", Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '09), New York, NY, USA, pp. 756-757, 2009.

[17] W. X. Zhao and et al., "Topical keyphrase extraction from Twitter", Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 379-388, 2011.

[18] Z. Liu, P. Li, Y. Zheng, and M. Sun, "Clustering to find exemplar terms for keyphrase extraction", Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP '09), vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 257-266, 2009.

[19] Z. Liu, P. Li, Y. Zheng, and M. Sun, "Automatic keyphrase extraction via topic decomposition", Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10), Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 366-376, 2010.