*ICIC Express Letters*
*Part B: Applications*
Volume **10**, Number **10**, October **2019**

ICIC International ©2019 ISSN 2185-2766
pp. 875–884

# A NOVEL APPROACH TO ARABIC KEYPHRASE EXTRACTION

Dhiaa Musleh[1], Rashad Ahmed[3], Atta-ur-Rahman[1,*] and Fahd Alhaidari[2]

[1]Department of Computer Science
[2]Department of Computer Information System
College of Computer Science and Information Technology
Imam Abdulrahman Bin Faisal University
P.O. Box 1982, Dammam 31441, Saudi Arabia
*Corresponding author: aaurrahman@iau.edu.sa

[3]Department of Information and Computer Science
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia

Received April 2019; accepted July 2019

Abstract. *Keyword extraction is one of the most important research areas of information retrieval. The task is challenging, and it has been receiving the attention of researchers in the last decade. The importance of this problem originates from the fact that extracted keywords can be used in many fields such as document indexing, clustering, classification, summarization, metadata generation, topic identification, and information visualization. In addition, recent years have witnessed a dramatic growth in the number of documents that are available online with no key-phrases assigned. Assigning keyphrase to such documents manually is impractical. This situation demands automatic keyphrase extraction. In this regard, several approaches have been proposed in the literature. These approaches use techniques borrowed from areas such as machine learning, computational linguistic and statistical analysis. In this paper, Arabic keyphrase extraction system is developed for Arabic documents. A new boosting factor is proposed by which occurrence of compound terms is boosted based on occurrences of their words. This is motivated by the fact that long phrases are preferred to be keywords than single words. The performance of the proposed keyphrase extraction method is evaluated using three Arabic datasets and the results show that the proposed method has comparable performance to that of KP-Miner.*
**Keywords:** Arabic keyphrase extraction, KP-Miner, Arabic documents

1. **Introduction.** In the last few years, there has been a trend to automatic keyphrase extraction from documents. There are at least two reasons for that. First, huge numbers of documents have been made available online; however, most of them do not have keywords. Manual assignment of keyphrase is time consuming, costly and error prone. Second, keyphrase extraction is useful for many applications such as document indexing, clustering, classification, summarization and retrieval. This trend has resulted in several researches addressing this problem. Prior research in this field has focused almost exclusively on the keyword extraction. However, in practice, keyphrases are also used to describe, index and summarize documents. Early ideas and techniques of automatically keyphrase extraction from documents date back into the nineties [1]. First approaches to tackle this problem used heuristics. However, they were unsuccessful to map well to those keywords assigned by authors. Motivated by this failing, there have been many different attempts to that goal, combining several major techniques such as computational linguistic techniques, machine learning techniques and statistical techniques, with the conclusion that while it is quite feasible to extract a set of keyphrases from documents, it is still very hard to produce such keyphrases if one aims applying it to general domain

DOI: 10.24507/icicelb.10.10.875

real data. Several models exist for keyphrase extraction. Those extraction approaches are based on machine learning, statistical analysis and computational linguistic techniques. In this paper, we propose a new boosting factor by which occurrences of compound terms are boosted based on the occurrences of their words. This is motivated by the fact that long phrases are preferred to be keywords than single words. The proposed work is based on KP-Miner, a technique proposed by El-Beltagy and Rafea [1], with some modifications. KP-Miner introduces a boosting factor for compound terms; their boosting factor is based on the view that the frequency of compound terms is much less than the frequency of single terms within the same document. However, this boosting factor increases all compound terms by the same ratio. The rest of this paper is organized as follows. Section 2 presents a background on keyphrase extraction task and gives a detailed description of the keyphrase extraction techniques proposed in the literature. The proposed work presented in Section 3. Section 4 presents a detailed discussion of the experimental work. The paper is concluded with summary in Section 5.

## 2. Background: Keyphrase Extraction.

Keyphrase extraction is relatively new field in natural language processing. Its main objective is to improve information retrieval. Potential applications of keyphrase extraction are document indexing, clustering, classification, summarization and retrieval. Keyphrases of a document are usually phrases that best reflect and describe the documents contents [2]. Keyphrase extraction is the process of identifying this set of phrases from a document. Two types of approaches for automatic keywords indexing are distinguished in the literature: keyword assignment and keyword extraction. Keyword assignment approaches select keywords from a predefined dictionary [3]. On the other hand, in keyword extraction the keywords are selected from the text based on words properties such as word frequency and word position. A general framework for keyword extraction mainly consists of three primary phases: selecting candidates to be keyphrases, assigning weight to each candidate and selecting the keyphrases with the highest weight. In the literature, three types of keyphrase extraction approaches are distinguished, namely machine learning, statistical and linguistic approaches.

### 2.1. Machine learning approaches.

Generally, machine learning approaches such as [4-7] make use of a trained model to extract keywords for new documents. The first technique which approaches a keyphrase extraction task as a supervised learning task is GenEx [7]. Another machine learning based approach is KEA (Keyphrase Extraction Algorithm) [8]. KEA makes use of a Bayesian learning model for keyphrase extraction task. In [6], Frank et al. extended KEA by making use of the likelihood of a particular phrase being as a keyphrase. Medelyan and Witten [9] proposed a new keyphrase indexing algorithm that combines the advantages of keyphrase extraction and keyphrase assignment approaches. An algorithm for keyphrase extraction based on combinations of a thesaurus and the frequency analysis using machine learning algorithm and morphological preprocessing tools was proposed in [10]. HaCohen-Kerner et al. [11,12] proposed a keyphrase extraction technique in which baseline methods were combined using supervised machine learning. An SOM (Self-Organizing Map) based approach to keyphrase extraction has been presented in [13]. The SOM (Self-Organizing Map) has been trained to classify a candidate phrase as keyphrase or not. Wang et al. [14,15] proposed a neural network based model to extract keyphrase from documents. In this model, keyphrase extraction has been viewed as a crisp binary classification task. Barker and Cornacchia [16] proposed an algorithm to select noun phrases as keyphrases for a document. The algorithm selects noun phrases from a document using a base noun phrase skimmer and an off-the-shelf online dictionary. Sarkar et al. [17] presented a novel keyphrase extraction approach using neural networks. This approach is based on the estimated class probabilities as the confidence scores which are used in re-ranking the phrases belonging to a class. An unsupervised keyphrase extraction algorithm for single Arabic documents called AKEA is proposed in [31]. The proposed algorithm relies on heuristics that collaborate linguistic patterns based on Part-of-Speech (POS) tags, statistical knowledge and the internal structural pattern of terms (i.e., word-occurrence). Artificial Neural Network (ANN) is used in [34] with word feature for keyword extraction. The reported performance was 0.83 in terms of a $G$ mean and 0.96 in terms of $f$-measure.

2.2. **Statistical approaches.** Statistical keyphrase extraction plays an important role in information retrieval, with many applications in data-mining and text classification systems [16]. Early ideas and techniques of automatically extracting keyphrases from documents using the frequency occurrences of words are proposed by Sarkar et al. [17]. This was followed by many refinements and developments. These techniques are simple and do not require the training data [18]. The statistical information of the words is used to extract keywords from documents. Several keyword extraction techniques [11,19-21] make use of document information such as the structure of documents, number of occurrences of words, and the co-occurrences of terms. Simple keyword extraction is based on phrase frequency such as in [3] while complex ones make use of statistical techniques such as in [20]. Term Frequency and Inverted Document Frequency (TF-IDF) weighting has been commonly used for keyphrase extraction. This is based on the view that terms with the highest frequency are most likely to be keywords. A huge amount of work has shown that TF-IDF is very useful in extracting keywords for scientific journals [22]. Li et al. [3] proposed algorithm for keyword extraction based on TF-IDF. This algorithm selects candidate keywords of unigrams, bigrams, trigrams based on features defined according to morphological character and context information. Matsuo and Ishizuka [20] presented a co-occurrence statistical based method using a clustering approach to extract keywords for documents without using a corpus. Co-occurrence indicates how many times two terms occur together within the same document. The AKE [23] is a keyword extraction system which is designed to extract keywords from news article. AKE makes use of existing statistical and linguistic techniques. Xie et al. [29] developed an $n$-gram based keyphrases extraction system where a sequential pattern mining based extraction method is proposed. The method can capture semantic relationships between words, and is able to discover different types of sequential patterns as candidates for keyphrase extraction. A method for Bangla language based on pronoun replacement and sentence ranking is proposed in [32]. The proposed method uses a rule-based system, hidden Markov model and Markov chain model.

2.3. **Computational linguistic approaches.** In general, linguistic based techniques start to run after a first statistical analysis. These approaches utilize the linguistic features of the word, sentences and documents. The first study on using the linguistic features for keyword extraction was done by Hulth [22]. Some of the linguistic information is noun phrases, predefined Part-of-Speech (POS) tags. In addition to those features, linguistic features were used to improve the process of phrase candidate selection. In [24], Breiman experimented bagging to train the system and the experimental results show that linguistic knowledge to the term representation improves keyword extraction accuracy. Ercan and Cicekli [25] proposed a keyword extraction technique based on lexical chain. Semantic content of text is represented using lexical chain which contains a subset of words in the text which are semantically related. In [26], three methods to extract keywords in open-domain multilingual textual resources were presented based on statistical associations and through enhanced semantic clustering. Akin to [35], semantic network is used for the same purpose. Avanzo et al. [27,28] proposed a new keyphrase extraction technique, namely LAKE, which stands for Learning Algorithm for Keyphrase. In [33], the authors proposed a new method called CFinder which combines statistical knowledge and domain-specific knowledge to indicate the importance of the terms within the domain. Rafiei-Asl and Nickabadi [30] proposed a Topical and Structural Automatic Keyphrase Extractor (TSAKE) that combines the prior knowledge about the input langue learned by an $n$-gram topical model (TNG) with the co-occurrence graph of the input text to form some topical graphs.

3. **Proposed Work.** The proposed work is based on KP-Miner, a technique proposed by El-Beltagy and Rafea [1], with some modifications. KP-Miner introduces a boosting factor for compound terms. Their boosting factor is based on the view that the frequencies of compound terms are much less than the frequencies of single terms within the same document. However, this boosting factor increases all compound terms by the same ratio. In our work, we propose a new boosting factor by which occurrences of compound terms are boosted based on the frequency of their words. This is motivated by the fact that long phrases are preferred to be keywords than single words. The general process of our work follows these steps.

1) **Construction of candidate keyphrase set:** This phase is composed of three steps which are, document preprocessing, building $n$-gram models and filtering $n$-grams.
   a) ***Document preprocessing:*** Before building $n$-gram models, documents were preprocessed to remove punctuation marks, diacritics, non-Arabic letters.
   b) ***Building n-gram models:*** In this step, all possible word $n$-grams are computed. Then, the frequency of occurrence of each $n$-gram is computed.
   c) ***Filtering n-grams:*** In this step, some of the $n$-grams are removed according to the following conditions. The first condition is that any $n$-grams occurring less than $n$ times in the document cannot be a candidate keyphrase. The second condition is that all $n$-grams that appear for a first time before a predefined threshold are likely to be keyphrases. The third condition is that a candidate keyphrase cannot start or end with stop words. Finally, all $n$-grams having a verb are ignored.

2) **Weight calculating of candidate keyphrases:** The following features are used to rank the candidate keyphrases: Term Frequency (TF), position of the phrase and the boosting factor. The final weight assigned to each candidate keyphrase is computed using the following equation.

$$W_i = newF_i * P_i * tf_i * idf \tag{1}$$

where $W_i$ is the weight of term $i$ in Document $D$, $tf_i$ is the frequency of term $i$ in the same document, $P_i$ is the term position of term $i$, $newF_i$ is the boosting factor and $idf$ is $\log_2(N/n)$, where $N$ is the number of documents in the collection and $n$ is the number of documents where term $i$ occurs at least once. The boosting factor for bi-gram is defined as:

$$newF_i = \alpha_1[count(w_1) + count(w_2)] + \alpha_2 count(w_1w_2) \tag{2}$$

The boosting factor for tri-gram is defined as

$$newF_i = \alpha_1[count(w_1w_2) + count(w_2w_3)] + \alpha_2 count(w_1w_2w_3) \text{ where } \alpha_1 + \alpha_2 = 1 \tag{3}$$

The values for $\alpha_1$ and $\alpha_2$ are selected experimentally.

3) **Final keyphrase selection and post processing:** Candidate keyphrases are ranked according to the calculated weights. The top $n$ candidate keyphrase are considered as keyphrases.

The final list of keyphrases will be generated by ranking the candidate keyphrases according to their calculated weights. Then, user can select from the list the desired number of keyphrases, for example, 10 keyphrases, which will be the top 10 keyphrases from the list in this case. Unlike many keyword extracting approaches, the proposed approach does not require a training phase on a specific dataset in order to perform its task.

4. **Experimental Work.** To evaluate the performance of the developed keyword extraction approach, we tested it using three different datasets collected from three different sources and containing a total of 50 documents. The characteristics of used data sets are presented in Section 4.1. To evaluate how well the developed keyword extraction approach performs, we compare the keywords extracted by developed system with those author-assigned keywords in the corresponding documents. Then, results were compared to the keywords extracted by KP-Miner for the same documents. We selected KP-Miner to compare with for two reasons. Firstly, KP-Miner is efficient and effective in extracting keywords from documents written in English or Arabic language [1]. Secondly, KP-Miner has an online tool to automatically extract keywords so it can be evaluated on the same datasets that we used with the developed system.

4.1. **Research data.** In our experiments, three different datasets were collected: the Jordanian Journal of Educational Sciences (JJES) (20 documents), Damascus University Journal for Basic Sciences (20 documents), Zarqa Journal for Research and Studies in Humanities (10 documents). The following table shows the characteristics of the datasets.

| **Dataset** | **Source** | **Domain** | **Docs** |
|---|---|---|---|
| **1** | The Jordanian Journal of Educational Sciences | Educational Sciences | 20 |
| **2** | Damascus University Journal for Basic Sciences | Basic Sciences | 20 |
| **3** | Zarqa Journal for Research and Studies in Humanities | Social Science | 10 |

4.2. **Experimental results.** In this section, we discuss the results of our experiments. We have three datasets, and for each set we run two experiments. In the first experiment, $N$ keywords were extracted from the document by using both the proposed approach and KP-Miner where $N$ represents the number of author-assigned keywords for the corresponding document. In the second experiment, 10 keywords were extracted from the document by using both the proposed approach and KP-Miner. In both experiments, the accuracy of each system was computed based on the number of relevant extracted keywords and $N$ as follows:

$$\text{Accuracy} = \frac{\text{number of relevant extracted keywords}}{\text{number of author assigned keywords}}$$

Table 1 shows the experimental results for each document in Dataset 1. The average of all documents is also shown in Table 1. It is obvious from the table that the accuracy when we used top 10 extracted keywords is better than using top $N$ for both systems. This is expected since $N$ is always less than 10 which means by extracting 10 keywords the probability of finding relevant keywords will be higher. Figure 1 shows the result for the top 10 extracted keywords. From Table 1 and Figure 1, we can see that the proposed approach outperforms KP-Miner in this dataset.

Table 2 and Figure 2 show the experimental results for Dataset 2.

As the previous one, the accuracy when we used top 10 extracted keywords is better than using top $N$ for both systems. Also, the proposed approach outperforms KP-Miner for both top $N$ and top 10 extracted keywords. Table 3 and Figure 3 show the experimental results for each document and the average for Dataset 3. The table shows comparable results between our approach and KP-Miner where KP-Miner performs slightly better than our approach for top $N$ keyphrases. The proposed approach still outperforms KP-Miner for top 10 keyphrase extraction.

Table 1. Experimental results for Dataset 1

| Doc ID | $N$ | Proposed Approach | | | | KP-Miner | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Top $N$ | Accuracy | Top 10 | Accuracy | Top $N$ | Accuracy | Top 10 | Accuracy |
| 1 | 5 | 0 | 0 | 1 | 0.2 | 0 | 0 | 0 | 0 |
| 2 | 4 | 1 | 0.25 | 1 | 0.25 | 0 | 0 | 1 | 0.25 |
| 3 | 5 | 1 | 0.2 | 2 | 0.4 | 1 | 0.2 | 2 | 0.4 |
| 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 2 | 1 | 0.5 | 1 | 0.5 | 1 | 0.5 | 1 | 0.5 |
| 6 | 4 | 2 | 0.5 | 3 | 0.75 | 2 | 0.5 | 2 | 0.5 |
| 7 | 4 | 1 | 0.25 | 2 | 0.5 | 0 | 0 | 2 | 0.5 |
| 8 | 4 | 1 | 0.25 | 1 | 0.25 | 0 | 0 | 1 | 0.25 |
| 9 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 3 | 2 | 0.66 | 2 | 0.66 | 0 | 0 | 1 | 0.33 |
| 11 | 3 | 1 | 0.33 | 1 | 0.33 | 0 | 0 | 0 | 0 |
| 12 | 4 | 1 | 0.25 | 1 | 0.25 | 0 | 0 | 1 | 0.25 |
| 13 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 4 | 0 | 0 | 1 | 0.25 | 0 | 0 | 1 | 0.25 |
| 15 | 3 | 2 | 0.66 | 3 | 1 | 0 | 0 | 1 | 0.33 |
| 16 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.25 |
| 17 | 4 | 0 | 0 | 1 | 0.25 | 0 | 0 | 0 | 0 |
| 18 | 2 | 0 | 0 | 1 | 0.5 | 1 | 0.5 | 1 | 0.5 |
| 19 | 4 | 1 | 0.25 | 2 | 0.5 | 0 | 0 | 1 | 0.25 |
| 20 | 3 | 1 | 0.33 | 2 | 0.66 | 1 | 0.33 | 1 | 0.33 |
| **Total** | **72** | **15** | | **25** | | **6** | | **17** | |
| **Average** | | | **0.21** | | **0.35** | | **0.08** | | **0.24** |

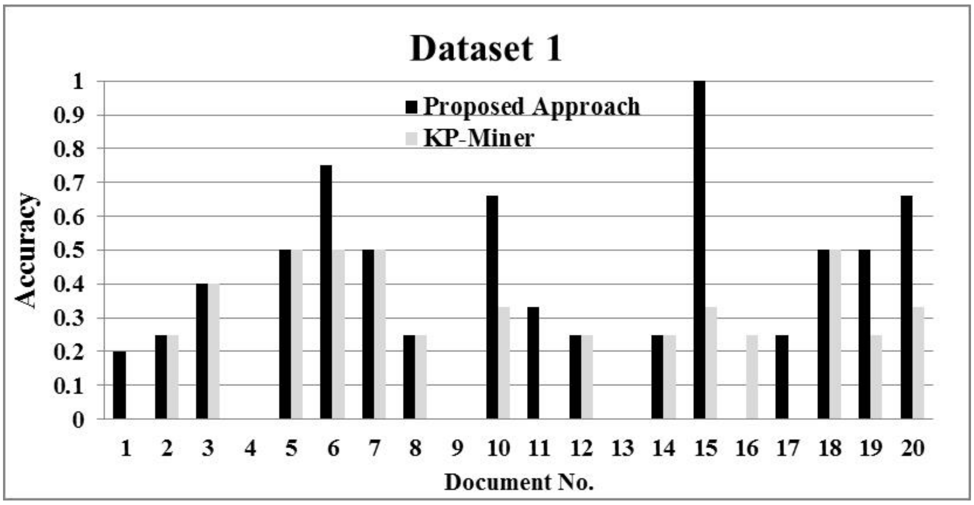

Figure 1. Experimental results for Dataset 1

4.3. **Result analysis.** Considering the above results, there is a gap between extracted keyphrases by both KP-Miner and the proposed technique and the keyphrases assigned by authors. This can be contributed to many reasons. One reason is that some authors assigned keywords that are not representative of the documents. Also, we notice that some documents have only one author assigned keyphrase and in some of these documents have single word author assigned keywords. Others have assigned keyphrases whose single words can be considered as stop words such as "شمال غرب سورية". The previous keyphrase is assigned by the author to a document whereas the first two words are considered as stop words. In general, the data used to train and test keyphrase extraction models

Table 2. Experimental results for Dataset 2

| **Doc ID** | ***N*** | **Proposed Approach** | | | | **KP-Miner** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **Top *N*** | **Accuracy** | **Top 10** | **Accuracy** | **Top *N*** | **Accuracy** | **Top 10** | **Accuracy** |
| 1 | 3 | 2 | 0.67 | 3 | 1 | 1 | 0.33 | 2 | 0.67 |
| 2 | 3 | 0 | 0 | 1 | 0.33 | 0 | 0 | 0 | 0 |
| 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 5 | 1 | 0.2 | 1 | 0.2 | 1 | 0.2 | 1 | 0.2 |
| 5 | 3 | 1 | 0.33 | 1 | 0.33 | 0 | 0 | 0 | 0 |
| 6 | 9 | 0 | 0 | 0 | 0 | 1 | 0.11 | 1 | 0.11 |
| 7 | 3 | 1 | 0.33 | 1 | 0.33 | 0 | 0 | 1 | 0.33 |
| 8 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 5 | 1 | 0.2 | 1 | 0.2 | 1 | 0.2 | 1 | 0.2 |
| 10 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 10 | 1 | 0.1 | 1 | 0.1 | 2 | 0.2 | 2 | 0.2 |
| 12 | 5 | 2 | 0.4 | 2 | 0.4 | 1 | 0.2 | 1 | 0.2 |
| 13 | 4 | 1 | 0.25 | 1 | 0.25 | 1 | 0.25 | 1 | 0.25 |
| 14 | 7 | 1 | 0.14 | 1 | 0.14 | 0 | 0 | 0 | 0 |
| 15 | 6 | 3 | 0.5 | 3 | 0.5 | 1 | 0.17 | 2 | 0.33 |
| 16 | 5 | 1 | 0.2 | 2 | 0.4 | 1 | 0.2 | 1 | 0.2 |
| 17 | 9 | 2 | 0.22 | 2 | 0.22 | 1 | 0.11 | 1 | 0.11 |
| 18 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | 6 | 2 | 0.33 | 2 | 0.33 | 1 | 0.17 | 2 | 0.33 |
| 20 | 4 | 0 | 0 | 1 | 0.25 | 1 | 0.25 | 1 | 0.25 |
| **Total** | **67** | **14** | | **16** | | **10** | | **12** | |
| **Average** | | | **0.21** | | **0.24** | | **0.15** | | **0.18** |

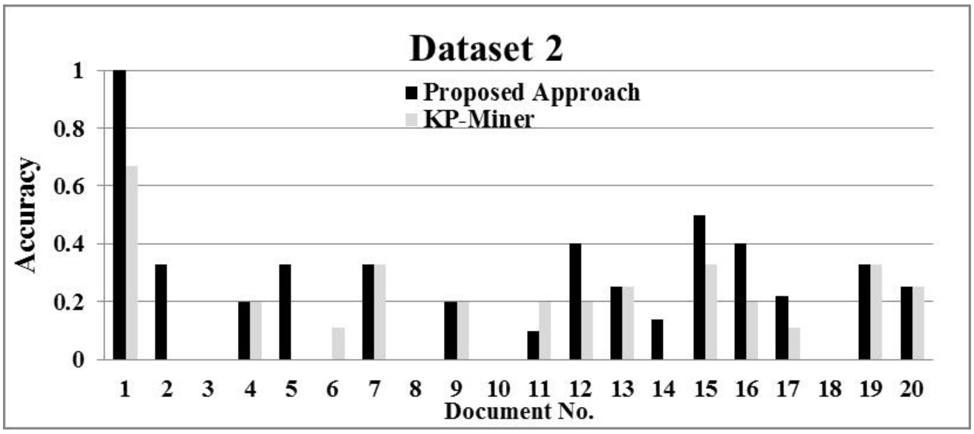

Figure 2. Experimental results for Dataset 2

have great impacts on their performance. In addition to above, preprocessing steps such as stemming, and part-of-speech tagging affect the performance of keyphrase extraction techniques to some extent. The experimental results show that the proposed approach, which implements the new boosting factor, outperforms KP-Miner approach in Dataset 1 and Dataset 2 for both top $N$ and top 10 keyphrases. For Dataset 3, KP-Miner performs slightly better than the proposed approach for top $N$ keyphrases whereas the proposed approach still outperforms KP-Miner for top 10 keyphrases extraction. Tables 4 and 5 show the experimental results for all datasets with top $N$ and top 10 extracted keyphrases respectively.

5. **Conclusion.** There is a significant amount of work concerning keyphrase extraction. Proposed keyword extraction approaches in the literature use techniques borrowed from

Table 3. Experimental results for Dataset 3

| Doc ID | $N$ | Proposed Approach | | | | KP-Miner | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Top $N$ | Accuracy | Top 10 | Accuracy | Top $N$ | Accuracy | Top 10 | Accuracy |
| 1 | 7 | 1 | 0.14 | 1 | 0.14 | 1 | 0.14 | 1 | 0.14 |
| 2 | 5 | 0 | 0 | 1 | 0.2 | 1 | 0.2 | 1 | 0.2 |
| 3 | 7 | 2 | 0.29 | 2 | 0.29 | 2 | 0.29 | 2 | 0.29 |
| 4 | 3 | 0 | 0 | 2 | 0.67 | 1 | 0.33 | 1 | 0.33 |
| 5 | 3 | 2 | 0.67 | 2 | 0.67 | 1 | 0.33 | 2 | 0.67 |
| 6 | 3 | 2 | 0.67 | 2 | 0.67 | 1 | 0.33 | 1 | 0.33 |
| 7 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 4 | 0 | 0 | 1 | 0.25 | 0 | 0 | 0 | 0 |
| 9 | 4 | 0 | 0 | 1 | 0.25 | 0 | 0 | 0 | 0 |
| 10 | 3 | 0 | 0 | 1 | 0.33 | 1 | 0.33 | 1 | 0.33 |
| **Total** | **43** | **7** | | **13** | | **8** | | **9** | |
| **Average** | | | **0.16** | | **0.30** | | **0.19** | | **0.21** |

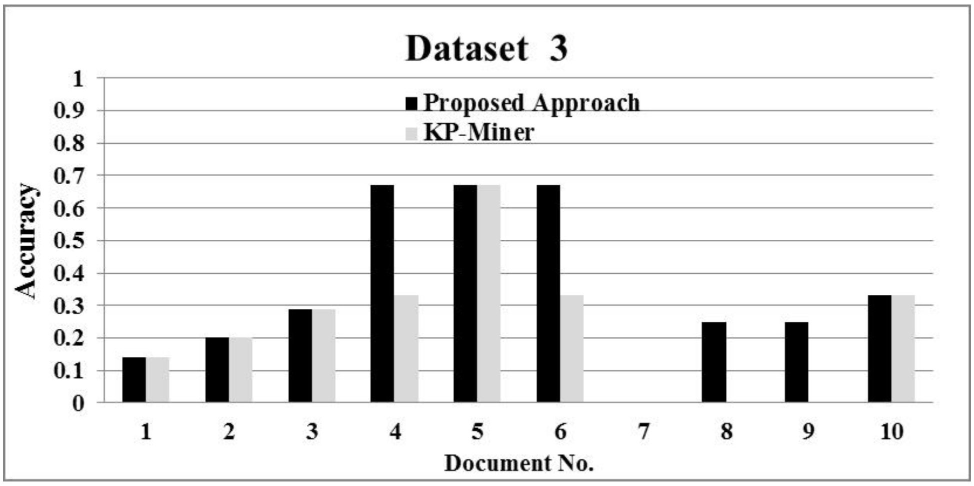

Figure 3. Experimental results for Dataset 3

Table 4. Experimental results for all datasets with top $N$* extracted keyphrases

| Dataset | #Docs. | #Author Keywords | Proposed Approach | | KP-Miner | |
|---|---|---|---|---|---|---|
| | | | Key | Average | Key | Average |
| **1** | 20 | 72 | 15 | 0.21 | 6 | 0.08 |
| **2** | 20 | 67 | 14 | 0.21 | 10 | 0.15 |
| **3** | 10 | 43 | 7 | 0.16 | 8 | 0.19 |
| *$N$ represents number of authors assigned keyphrases | | | | | | |

Table 5. Experimental results for all datasets with top 10 extracted keyphrases

| Dataset | #Docs. | # Author Keywords | Proposed Approach | | KP-Miner | |
|---|---|---|---|---|---|---|
| | | | Key | Average | Key | Average |
| **1** | 20 | 72 | 25 | 0.35 | 17 | 0.24 |
| **2** | 20 | 67 | 16 | 0.24 | 12 | 0.18 |
| **3** | 10 | 43 | 13 | 0.30 | 9 | 0.21 |

areas such as computational linguistic, machine learning and statistical analysis. In this paper, a keyphrase extraction approach is proposed using a new boosting factor by which occurrences of compound terms are boosted based on the frequency of their words. This is motivated by the fact that long phrases are preferred to be keywords than single words. The proposed approach was tested using three different databases and the results were compared with KP-Miner approach. The experimental results show that the proposed approach outperforms KP-miner approach with two databases, and it gives a comparable performance to that of KP-Miner with the third database. One possible direction for future work for keyphrase extraction problem is to take words synonym into consideration when ranking words.

## REFERENCES

[1] S. R. El-Beltagy and A. Rafea, KP-Miner: A keyphrase extraction system for English and Arabic documents, *Information Systems*, vol.34, no.1, pp.132-144, 2009.

[2] Q. Zhang and C. Zhang, Automatic Chinese keyword extraction based on KNN for implicit subject extraction, *International Symposium on Knowledge Acquisition and Modeling*, pp.689-692, 2008.

[3] J. Li, Q. Fan and K. Zhang, Keyword extraction based on tf/idf for Chinese news document, *Wuhan University Journal of Natural Sciences*, vol.12, no.5, pp.917-921, 2007.

[4] O. Medelyan and H. Witten, *Thesaurus-Based Index Term Extraction for Agricultural Documents*, 2005.

[5] K. Zhang, H. Xu, J. Tang and J. Li, Keyword extraction using support vector machine, *International Conference on Web-Age Information Management*, pp.85-96, 2006.

[6] E. Frank, W. Gordon, W. Paynter, I. Witten, C. Gutwin and C. Nevill-Manning, Domain-specific keyphrase extraction, *The 16th International Joint Conference on Artificial Intelligence (IJCAI 99)*, San Francisco, CA, USA, vol.2, pp.668-673, 1999.

[7] P. Turney, Learning to extract keyphrases from text, *CoRR*, 1999.

[8] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin and C. G. Nevill-Manning, KEA: Practical automated keyphrase extraction, *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, pp.129-152, 2005.

[9] O. Medelyan and I. H. Witten, Thesaurus based automatic keyphrase indexing, *Machine Learning*, pp.6-7, 2006.

[10] A. Hulth, J. Karlgren, A. Jonsson, H. Boström and L. Asker, Automatic keyword extraction using domain knowledge, *International Conference on Intelligent Text Processing and Computational Linguistics*, pp.472-482, 2001.

[11] Y. HaCohen-Kerner, Automatic extraction of keywords from abstracts, *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp.843-849, 2003.

[12] Y. HaCohen-Kerner, Z. Gross and A. Masa, Automatic extraction and learning of keyphrases from scientific articles, *International Conference on Intelligent Text Processing and Computational Linguistics*, pp.657-669, 2005.

[13] A. P. Azcarraga and T. Yap Jr, Comparing keyword extraction techniques for WEBSOM text archives, *International Journal on Artificial Intelligence Tools*, vol.11, no.2, pp.219-232, 2002.

[14] T. Jo, Neural based approach to keyword extraction from documents, *International Conference on Computational Science and Its Applications*, pp.456-461, 2003.

[15] J. Wang, H. Peng and J. Hu, Automatic keyphrases extraction from document using neural network, *Advances in Machine Learning and Cybernetics*, pp.633-641, 2006.

[16] K. Barker and N. Cornacchia, Using noun phrase heads to extract document keyphrases, *Conference of the Canadian Society for Computational Studies of Intelligence*, pp.40-52, 2000.

[17] K. Sarkar, M. Nasipuri and S. Ghose, A new approach to keyphrase extraction using neural networks, *Journal of Computer Science*, vol.7, no.2, 2010.

[18] C. Zhang, H. Wang, Y. Liu, D. Wu, Y. Liao and B. Wang, Automatic keyword extraction from documents using conditional random fields, *Journal of Computational Information Systems*, vol.4, no.3, pp.1169-1180, 2008.

[19] F. Fukumoto, Y. Suzukit and J. Fukumoto, An automatic extraction of key paragraphs based on context dependency, *Proc. of the 5th Conference on Applied Natural Language Processing*, pp.291-298, 1997.

[20] Y. Matsuo and M. Ishizuka, Keyword extraction from a single document using word co-occurrence statistical information, *International Journal on Artificial Intelligence Tools*, vol.13, no.1, pp.157-169, 2004.

[21] M. Rossignol and S. Pascale, Combining statistical data analysis techniques to extract topical keyword classes from corpora, *Intelligent Data Analysis*, vol.9, pp.105-127, 2005.

[22] A. Hulth, Improved automatic keyword extraction given more linguistic knowledge, *Proc. of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp.216-223, 2003.

[23] J. L. Martínez-Fernández, A. García-Serrano, P. Martínez and J. Villena, Automatic keyword extraction for news finder, *International Workshop on Adaptive Multimedia Retrieval*, pp.99-119, 2003.

[24] L. Breiman, Bagging predictors, *Machine Learning*, vol.24, no.2, pp.123-140, 1996.

[25] G. Ercan and I. Cicekli, Using lexical chains for keyword extraction, *Information Processing & Management*, vol.43, no.6, pp.1705-1714, 2007.

[26] A. Panunzi, M. Fabbri and M. Moneglia, Keyword extraction in open-domain multilingual textual resources, *The 1st International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS'05)*, pp.253-256, 2005.

[27] E. D'Avanzo, B. Magnini and A. Vallin, Keyphrase extraction for summarization purposes: The LAKE system at DUC-2004, *Proc. of the 2004 Document Understanding Conference*, 2004.

[28] E. D'Avanzo and B. Magnini, A keyphrase-based approach to summarization: The LAKE system at DUC-2005, *Proc. of the 2005 Document Understanding Conference*, 2005.

[29] F. Xie, X. Wu and X. Zhu, Efficient sequential pattern mining with wildcards for keyphrase extraction, *Knowledge-Based Systems*, vol.115, pp.27-39, 2017.

[30] J. Rafiei-Asl and A. Nickabadi, TSAKE: A topical and structural automatic keyphrase extractor, *Applied Soft Computing*, vol.58, pp.620-630, 2017.

[31] E. Amer and K. Foad, AKEA: An Arabic keyphrase extraction algorithm, *Proc. of the International Conference on Advanced Intelligent Systems and Informatics*, pp.137-146, 2016.

[32] M. M. Haque, S. Pervin and Z. Begum, An innovative approach of Bangla text summarization by introducing pronoun replacement and improved sentence ranking, *Journal of Information Processing Systems*, vol.13, no.4, pp.752-777, 2017.

[33] Y.-B. Kang, P. D. Haghighi and F. Burstein, CFinder: An intelligent key concept finder from text for ontology development, *Expert Systems with Applications*, vol.41, no.9, pp.4494-4504, 2014.

[34] A. P. Azcarraga, P. Tensuan and R. Setiono, Tagging documents using neural networks based on local word features, *International Joint Conference on Neural Networks (IJCNN)*, Beijing, 2014.

[35] Atta-ur-Rahman, Knowledge representation: A semantic network approach, in *Handbook of Research on Computational Intelligence Applications in Bioinformatics*, S. Dash and B. Subudhi (eds.), IGI Global, 2016.