

An Efficient Approach for Keyphrase Extraction from English Document

Imtiaz Hossain Emu, Asraf Uddin Ahmed

Department of Information and Communication Technology, Mawlana Bhashani Science and Technology University, Tangail, Bangladesh.
E-mail: {imtiaz.emu.it, 13ratul}@gmail.com

Md. Manowarul Islam, Md. Selim Al Mamun

Department of Electrical and Communication Engineering, Okayama University, Okayama, Japan.
E-mail: mano.cse.du@gmail.com, {manowar.islam, mamun0013}@s.okayama-u.ac.jp

Md. Ashraf Uddin

Department of Information Technology, Federation University, Australia.
E-mail: ashraf.cse.jnu@gmail.com, mdashrafuddin@students.federation.edu.au

Received: 09 April 2017; Accepted: 13 September 2017; Published: 08 December 2017

Abstract—Keyphrases are set of words that reflect the main topic of interest of a document. It plays vital roles in document summarization, text mining, and retrieval of web contents. As it is closely related to a document, it reflects the contents of the document and acts as indices for a given document. Extracting the ideal keyphrases is important to understand the main contents of the document. In this work, we present a keyphrase extraction method that efficiently finds the keywords from English documents. The methods use some important features of the document such as TF, TF*IDF, GF, GF*IDF, TF*GF*IDF for the purpose. Finally, the performance of the proposal is evaluated using well-known document corpus.

Index Terms—Keyphrase, Stemming, Keyphrase Nomination, Term Frequency, Inverse Document Frequency.

I. INTRODUCTION

Nowadays information is one of the most powerful and important weapons in the modern world. Every moment we are getting an increasingly large amount of data or information from various sources like emails, web pages, electronic documents, etc. But all the sources cannot fulfill user's expectation of the readers since it is more difficult to find the appropriate information from a huge amount of document. It is very much difficult for a human being to find out the summary or extracting the main topics from a large body of text for the very fast growing information. Automatic keyword extraction provides an efficient and effective way to summarize from large documents.

Keyphrases are the words that can easily extract the main issues or topics covered in a text document [1-4]. It

is very useful for classifying, clustering, summarizing text documents in an effective manner [5]. By finding the most specific words from a text, keyword extraction helps the reader to discover the quick overview about documents. So, Keywords provides a summary for a document which leads to improved information retrieval process [5-7]. Keyphrases extraction is necessary and important tools for many reasons [8]. For example, i) keyphrases provide a summarization which helps the readers to make quicker decision whether the given article worth in-depth reading, ii) They improve document indexing efficiency, iii) Enable reader to quickly find an article relevant to a specific subjects or issues and iv) Enable a search engine to make the search more precise for readers.

Many research works have been found in the literature to perform the keyphrases extraction from a large document. Some of them use domain-specific knowledge and training sets [4] [9] [10]. These approaches needed human interaction and need an application-specific knowledge. Graph-based text summarization is another effective and popular field of research [11]. In the graph-based model, various approaches and methodologies have been found to find the links between the sentences and the keyphrases. Links between the sentences build a graph that produces the shortest path from the first sentence to the last [12].

In this research work, we propose an unsupervised approach to find the keyphrases from a text document. In our proposed system, for an input document, we nominate the keyphrases from the document. To do this, we split the keywords in their root form using a modified stemmer. Then, we check the split words to be a noun or not and we consider the word for extraction for a noun. We also count all noun of the previous non-pronoun occurring sentence. We then use various weight measures such as TF (Term Frequency), GF (Global Frequency),

IDF (Inverse Document Frequency) for an increase or decrease the weight of those nominated keyphrases. In the last step, we evaluate some keyphrases with highest weight measure. After extracting those keyphrases for representing the document in shorter form, we compare our system generated output for a specific document with the original output of that document.

The rest of this paper is organized as follows. We describe related works and motivation in Section II and describe the corpus in section III. Our proposed method is presented in Section IV and the simulation results are presented in Section V. Finally, we conclude the paper in Section VI along with future research direction.

II. RELATED WORKS

To find the keyphrases from a large document, a lot of research works are found in the literature. By focusing on non-linguistic features many statistical approaches have been proposed. They use some terms like term frequency (TF), inverse document frequency (IDF), and position of a keyword. Many of works suggested that TF*IDF weight measure is an efficient and popular way for finding keywords [4] [8]. TF*IDF identifies keywords which appear frequently in the document. Some linguistic approaches and methods are found that uses linguistic features such as part-of-speech, syntactic structure. For a single document, a co-occurrence distribution based method has been found which used a clustering strategy for extracting keywords [3].

Many machine learning based mechanisms are found that use training document for Keyword extraction [13-14]. A large number of training documents is used as training set and based on the gain knowledge from the training set is used to find keywords from new documents. For example, KEA is a keyphrase extraction system, uses the Bayesian learning technique for keyphrase extraction task [9]. In KEA, for example, numbers, punctuation marks, dashes, and brackets etc are used as phrase boundaries and the candidate keyphrases are separated by splitting the input text document according to phrase boundaries.

Graph-based text summarization is another effective and popular field of research [11][15-17]. In the graph-based model, various approaches and methodologies have been found to find the links between the sentences and the keyphrases. Links between the sentences are used for building a graph and the shortest path from the first sentence to the last is considered [12]. Another method is proposed which is known as PAT-tree-based keyphrases extraction system for Chinese and other oriental languages [18]. Another graph-based approach was proposed that uses HITS (Hyperlink-Induced Topic Search) algorithm for sentence ranking and finds the keywords [19].

Many research works used Neural Network for keyphrases extraction. For example, authors in [20][19] exploit traditional term frequency, inverse document frequency, and position (binary) features. In another approach, authors provided a cluster-based model in

order to highlight parts of the text that are semantically related [21]. The clusters of sentences reflecting the topics of the document are analyzed to find the main topics. The web information is considered as a huge field for additional knowledge where Keyphrases can be extracted [22].

Many research articles have been found in the literature for keyphrase extraction used machine learning or data mining approaches [23-31]. A Naive Bayes approach is used for keyword generation [23]. Authors proposed an unsupervised noun phrases extraction method where noun phrases, their occurrence, and co-occurrences are used generate the keywords from text documents [24]. An N-gram based method was proposed in [25], where authors' generated paraphrases are based on trigrams approach. Another variation of n-gram approach called N-gram IDF is found where N-gram IDF enables to find and determine the dominant N-grams among overlapping ones and extract key terms of any length from texts [26]. A natural language processing based keyword generation method is proposed for term extraction in the agricultural domain [27][31].

III. DESCRIPTION OF THE CORPUS

Most of the keyphrases consist of one, two or three words. For the simulation of English documents, we use 'fao30' document corpus. This can be found from "<http://maui-indexer.blogspot.com/2009/10/data-sets-for-keyphrase-extraction-and.html>". Here, we do not create the keyphrases manually. The 'fao30' team created the possible extracted keywords. For simulation purpose, we compare the generated keyphrases with the predefined keyword generated by corpus.

IV. PROPOSED METHOD

The proposed methodology has three stages namely key phrase nomination, weight measurement and ranking the extracted keyphrases using important features and finally evaluate the keywords/keyphrases with highest weight measures. For the nomination phase, we split the keywords in their root form using a modified stemmer. Then we check the split words to be noun or not. If the word turns out to be a noun then we consider the word for extraction. In the second step, we use the data mining tools (weight measures) such as TF, GF, IDF for an increase or decrease the weight of those nominated keyphrases.

Finally, we evaluate some keyphrases with highest weight measure. The number of keyphrases to be extracted is maintained by a predefined threshold. After extracting those keyphrases for representing the document in shorter form, we compare our system generated output for a specific document with the original output of that document. Based on this comparison we measured the accuracy provide by our proposed system. Fig. 1 summarizes the propose approach. What follows, we describe each of the stages in details:

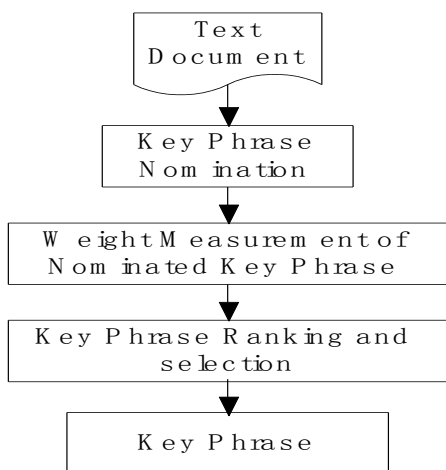


Fig.1. Overall design of proposed approach.

A. Key Phrase Nomination

Key Phrase nomination is the important stage to extract the keyphrases. Fig. 2 depicts the keyphrase nomination phase in details. Keyphrase nomination can be done by the following steps-

- a) Splitting Word
- b) Stemming and
- c) Noun Phrase Nomination.

Splitting Word: The first stage of keyphrase nomination is to splitting the words from the documents. It is also called stop word removal. In our approach we consider non-noun words which are previously POS tagged as our stop words. For English, we made a lexicon of more than 75000 non-noun word list which is considered as stop words. If the word from the document is found in these lists then it cannot be nominated for extraction. Only the noun or noun phrases are illegible for extraction [10]. We linked the pronoun with the nouns for the experiment. The result is based on both with pronoun and without the pronoun. After splitting word, we need to check split word is a noun or not, by checking a large database of the non-noun word for English. If the word does not exist in the database, then we considered that the word is a noun, otherwise not.

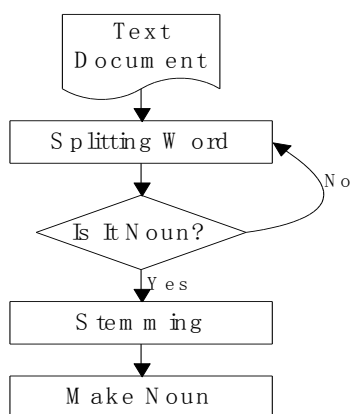


Fig.2. Key Phrase Nomination

Stemming: Stemming can be used to find the root term of a word. It was done by splitting the prefix and postfix from a word in a “longest match” basis using a predefined word list of prefix and postfix. In our approach, we only remove the postfixes [22]. Different stemming algorithms are available such as Lovin’s algorithm and Porter’s algorithm etc. The logic behind to select Porter is enlightened in Table 1.

Table 1. Comparison of different stemming algorithm

Word	Porter Stem	Lovin’s Stem	Iterated Lovins Stem
Believes	Believ	Belief	Belief
Belief	Belief	Belief	Belief
Believable	Believ	Belief	Belief
Jealousness	Jealous	Jeal	Jeal
Jealousy	Jealou	Jeal	Jeal
Police	Polic	Polic	Pol
Policy	Polici	Polic	Pol

Snowball algorithm is another algorithm that is the upgraded version of Porter’s algorithm. We have used the Snowball algorithm and modify it slightly mixing with some iteration to increase the stemmer’s performance. Suppose, the word ‘Nationalist’, according to Snowball it became ‘Nationalist’, then our iteration have made it ‘Nation’. We made a suffix list for finding the root word of a noun. We found the suffix list for English from various sites.

Noun Phrase Nomination: After stemming the keywords, we linked the keywords to one besides other to create NP chunks or Noun Phrases. We have considered both noun and noun phrases for the result. Then we find only the noun words from the corpus in root form. This nomination is done by measuring the term weights in different combinations. The noun which has greater weights was considered to be nominated. According to the weights, those nouns are considered and the noun phrases linked with them are nominated. The overall process is shown in Fig. 2.

B. Weight Measurement

A weight of the nominated keyphrases is measured using both local and global weighting features like term frequency measures (TF), inverse document frequency (IDF) etc., to the nominated keyphrases found in the previous stage. To be nominated for a final keyword, each of the keyphrases weight must be higher or equal to some predefined threshold value. The keyword with the highest weight locally is considered as a resulting keyword. Weights can be measured in two ways, locally and globally. The various weighting factors for local and global weight measurement of the keyword is described here.

Local weight: We use the following weight for our purpose.

- **TF:** Term Frequency is the number of times a

- phrase occurs in a document
- **Binary:** 1, if term appears in the document or 0, otherwise
- **log:** $\log_{10}(\text{TF})$

Global weight:

- **GF:** Global Frequency is measured for a particular word by using the following formula:

$$GF = \text{TF}_1 + \text{TF}_2 + \text{TF}_3 + \dots + \text{TF}_N \quad (1)$$

here N is equal to the number of documents.

- **DF:** Document frequency is defined as the no. of documents in which the term occurs.
- **IDF:** Inverse document frequency measure equation:

$$IDF = GF/DF. \quad (2)$$

or

$$IDF = \log_{10}(N/DF); \quad (3)$$

where N = No. of documents. We use IDF because it is optimal weight associated with a word feature in an information retrieval setting where we teach each document as the query that retrieves itself [12].

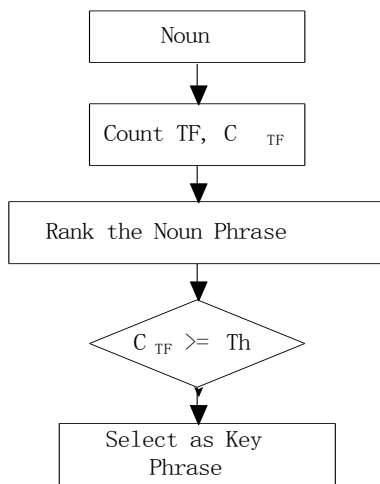


Fig.3. Ranking of nominated Keyphrases using TF

Frequency Measurement: We have used the combination of both local and global weights to rank the nominated words. If a word inside any Noun Phrase gets the highest weight then the Noun phrase also ranked high. We applied 5 combinations of frequency measurement. These are:

- TF measurement
- TF*IDF measurement
- GF measurement
- GF*IDF measurement
- TF*GF*IDF measurement

For measuring TF we did not use a database, because of it just based on the word which appears in the current document. We use “Counting Sort” with a hash table for faster measuring of TF shown in Fig 3.

For TF*IDF measurement we use the database for determining IDF. The multiplication of TF and IDF of each word is used for measuring weight. IDF value of each word is stored on database previously. Fig. 4 describes how we calculate the GF.

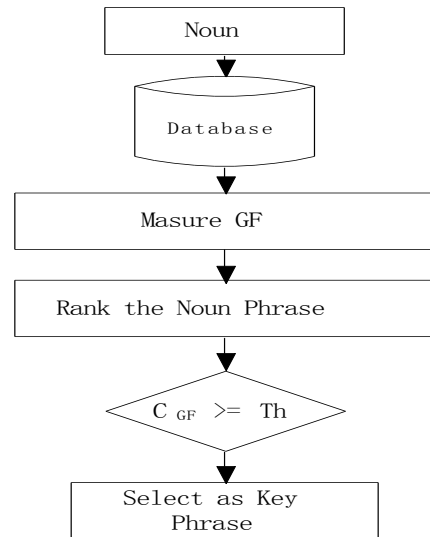


Fig.4. Ranking of nominated Keyphrases using GF

In case of TF*GF*IDF measurement, previously measured TF*IDF and GF*IDF are used and finally, a combination of the both is done to get TF*GF*IDF. The detail is shown in Fig. 5.

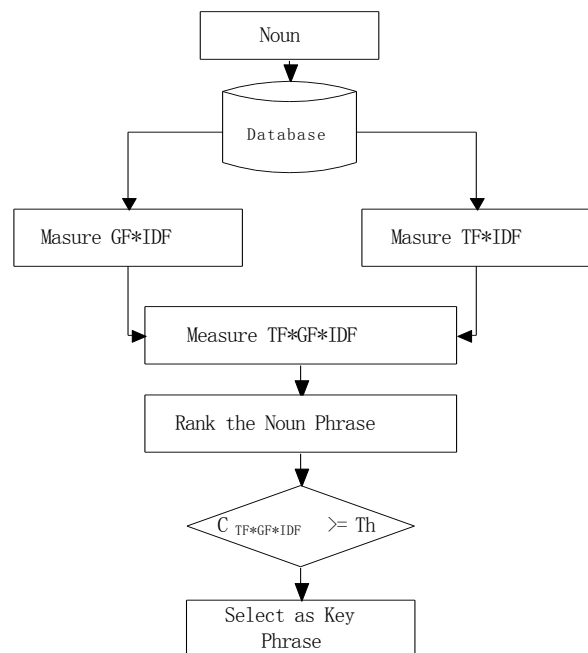


Fig.5. Ranking of nominated Keyphrases using TF*GF*IDF

C. Keyphrase Ranking and Selection

The score of a phrase k is computed based on the similarity of the phrase to the set of thematic terms in a document. The similarity of a phrase k to the set of thematic terms in a document is computed as the sum of the weight value of the thematic terms contained in the sentence k

$$S_k = \sum w_{t,k} \tag{4}$$

where $W_{t,k}$ is a local or global weight (such as TF, TF*IDF, GF, GF*IDF, TF*GF*IDF) of a thematic term t in a phrase k and S_k is the score of the phrase k .

The procedure of selecting the extracted keywords is based on the highest weights measured. We have a predefined threshold value for selecting top most phrases. If any term's weight is greater than or equal to the threshold then it has been selected for extraction. If any case of a tie, then we extract the keyword lexicographically.

V. PERFORMANCE EVALUATION

We evaluate the proposed prototype keyphrase extraction system using the two well-known matrices: Precision and Recall. These two matrices are combined to evaluate the results on the validation set using a formula, known as an F-Score measure.

$$F_B = \frac{(\beta^2 + 1) \times precision \times recall}{\beta^2 \times precision + recall} \tag{5}$$

Here, Precision indicates how many suggested keywords are correct? The formula for precision is [8]:

$$Precision = \frac{N}{K} \tag{6}$$

where N = number of keyphrases matched and K = number of keyphrases generated by the system.

On the other hand, Recall indicates many of the manually assigned keywords are found and of determined by the formula [8]:

$$Recall = \frac{N}{M} \tag{7}$$

where N = number of keyphrases matched and M = number of keyphrases generated manually.

The value of β was assigned to 1, thus giving Precision and Recall equal weights. After calculating the precision and recall for keyphrase extraction from each document we have calculated the average precision and recall (F-Score) for each of the above experiments. For each

experiment, the F-Score is generated for the combinations among local and global weights described in the previous section.

A. Result

For the simulation of English documents, we use 'fao30' document corpus. The 'fao30' team created the possible extracted keywords and we compare them with ours using precision and recall described in before and calculate the F-score. For more accurate result, we simulate our proposed approach many times.

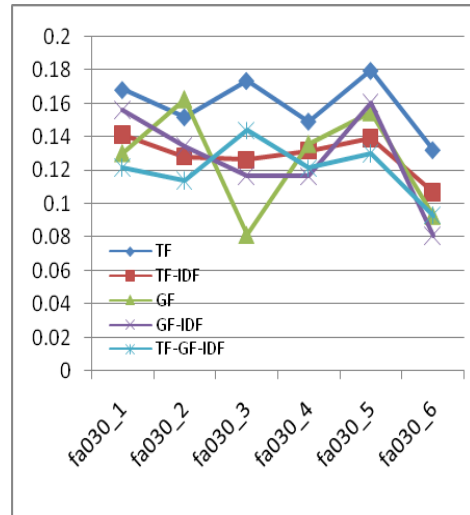


Fig.6. Weight measurement considering both noun and pronoun

For the average F-Score measure we consider the two facts, with a pronoun (P) and without pronoun (~P). The results of our proposed method, while considering noun and pronoun is given in Fig. 6. We found that TF provides the highest value than the others.

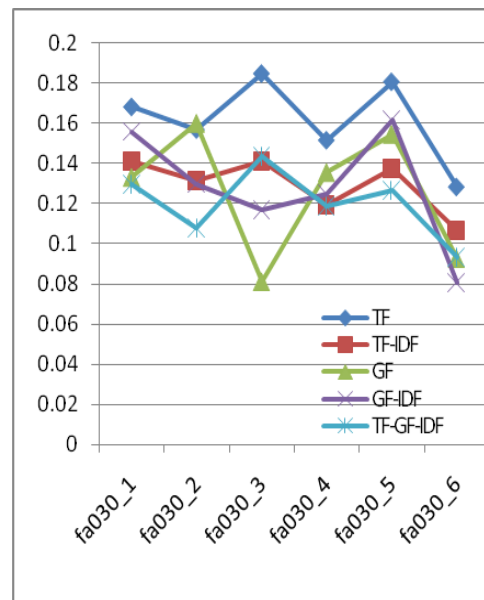


Fig.7. Weight measurement without pronoun

The result also shows that TF measure without pronoun is the best among the other four combinations

shown in Fig. 7. TF with pronoun measure is also good but in our approach, we linked the pronoun with all previous nouns. So, the accuracy level for this combination decreases.

Fig. 7 shows average F-score values of various local and global weights and TF outperforms the others. Table 2 shows the average F-score measure of local and global weights.

Table 2. Average F-score measure

TF(P)	0.159221
TF-IDF(P)	0.128775
GF(P)	0.126269
GF-IDF(P)	0.127458
TF-GF-IDF(P)	0.120692
TF(~P)	0.161872
TF-IDF(~P)	0.129561
GF(~P)	0.126377
GF-IDF(~P)	0.128384
TF-GF-IDF(~P)	0.120231

B. Discussion

In this paper, to measure the performance we used different combinations of some popular local and global weights for extracting keyphrases. We found that the TF measure without pronoun is the best among the other four combinations for English corpus. TF with pronoun measure is also good but in our approaches, we linked the pronoun with all previous nouns. If we consider the pronouns to be linked with only the proper nouns then the F-Score may be increased.

TF*IDF measurement is a well-known approach for extracting keywords. But from the results, we can clarify that if we consider only nouns/noun phrases to be extracted then TF is only needed to be measured. If we have to measure the IDF value, then we have to store the terms and it's weight. This will create a memory problem. Thus, in conclusion, we can say that in our proposed approach we can reduce the memory limit considering only TF value. This proposed prototype system works well for the small corpus.

VI. CONCLUSIONS

In this paper, we presented a keyphrase extraction method that uses combinations of local and global weights for extraction. The proposed method can extract the keyphrases in an efficient and effective way that depict the main concept or idea of the documents and provide a proper indication of the overall contents of the document.

The resulting analysis using different weighting factors shows that the TF method provides the best result than the other well-known keyphrase extraction weighting tools. The proposed method works well for small documents. In future, we will try to the moderation of our proposed works for more accuracy and for large documents as well.

REFERENCES

- [1] T. M. Froese B. Kosovac, D. J. Vanier. "Use of keyphrase extraction software for creation of an aec/fm thesaurus", *Journal of Information Technology in Construction*, pages 25–36, 2000.
- [2] M. Mahoui S.Jonse. "Hierarchical document clustering using automatically extracted keyphrase", *In proceedings of the third international Asian conference on digital libraries*, pages 113–120, Seoul, Korea, 2000.
- [3] Matsuo and M. Ishizuka. "Keyword extraction from a single document using word co-occurrence statistical information", *International Journal on Artificial Intelligence.*, 13(1):157–169, 2004.
- [4] A. Hulth. "Improved automatic keyword extraction given more linguistic knowledge", *In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223, Sapporo, Japan, 2003.
- [5] Q. Li Y. B. Wu. "Document keyphrases as subject metadata: incorporating document keys concepts in search results", *Journal of Information Retrieval.*, 11(13):229–249, 2008.
- [6] M. Staveley S. Jones. "Phrasier: A system for interactive document retrieval using keyphrases", *In Proceedings of SIGIR*, Berkeley, 1999.
- [7] C. Gutwin, G. Paynter, I. Witten, C. Nevill Manning, and E. Frank. "Document keyphrases as subject metadata: incorporating document keys concepts in search results", *Journal of Decision Support Systems.*, 27(1):81–104, 2003.
- [8] Kamal Sarkar. "Automatic keyphrase extraction from bengali documents: A preliminary study", *In Proceedings of Second International Conference on Emerging Applications of Information Technology*, India, 2011.
- [9] L. Plas, V.Pallotta, M.Rajman, and H.Ghorbel. "Automatic keyword extraction from spoken text. a comparison of two lexical resources: the edr and wordnet", *In Proceedings of the 4th International Language Resources and Evaluation, European Language Resource Association, 2004*, 2004.
- [10] I. H. Witten, G.W. Paynter, and E. Frank. "Kea: Practical automatic keyphrase extraction", *In Proceedings of Digital Libraries 99: The Fourth ACM Conference on Digital Libraries.*, pages 254–255, ACM Press, Berkeley, CA, 1999.
- [11] Y. Matsuo, Y. Ohsawa, and M. Ishizuka. "Keyword: Extracting keywords from a document as a small world", *In K. P. Jantke, A. shinohara (eds.): DS 2001. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg*, 2226(1):271–281, 2001.
- [12] Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. "Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction", *In ACL. The Association for Computational Linguistics*, 2007.
- [13] Y. HaCohen-Kerner. "Automatic extraction of keywords from abstracts", *In V. Palade, R. J. Howlett, L. C. Jain (eds.): KES 2003. Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin Heidelberg*, 2773:843–849, 2003.
- [14] Y. HaCohen-Kerner, Z. Gross, and A. Masa. "Automatic extraction and learning of keyphrases from scientific articles", *In A. Gelbukh (ed.): CICLing 2005. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg*, 3406:657–669, 2005.
- [15] Shi, Wei, Weiguo Zheng, Jeffrey Xu Yu, Hong Cheng, and Lei Zou. "Keyphrase Extraction Using Knowledge Graphs", *In Asia-Pacific Web (APWeb) and Web-Age*

- Information Management (WAIM) Joint Conference on Web and Big Data*, pp. 132-148. Springer, Cham, 2017.
- [16] Bougouin, Adrien, Florian Boudin, and B atrice Daille. "Keyphrase Annotation with Graph Co-Ranking", *arXiv preprint arXiv: 1611.02007*, 2016.
- [17] Murali Krishna V.V. Ravinuthala, Satyananda Reddy Ch., Thematic "Text Graph: A Text Representation Technique for Keyword Weighting in Extractive Summarization System", *International Journal of Information Engineering and Electronic Business(IJIEEB)*, Vol.8, No.4, pp.18-25, 2016. DOI: 10.5815/ijieeb.2016.04.03
- [18] Lee-Feng Chien. "Pat-tree-based adaptive keyphrase extraction for intelligent chinese information retrieval", *Inf. Process. Manage.*, 35(4):501– 521, 1999.
- [19] Marina Litvak. "Graph-based keyword extraction for single-document summarization", In *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24, 2008.
- [20] Jiabing Wang, Hong Peng, and Jing-Song Hu. "Automatic keyphrases extraction from document using neural network", In *ICMLC*, pages 633– 641, 2005.
- [21] Claude Pasquier. "Single document keyphrase extraction using sentence clustering and latent dirichlet allocation", In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval '10*, pages 154–157, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [22] P. D. Turney. "Learning algorithm for keyphrase extraction", *Journal of Information Retrieval*, 2(4):303–336, 2000.
- [23] Pabitha, P., Suganthi, S. and Ram, R.. "Key Phrase Extraction Using Naive Bayes' in Question Generation System", *Asian Journal of Information Technology*, 15(3), pp.372-375, 2016.
- [24] Kathait, S.S., Tiwari, S., Varshney, A. and Sharma, A. "Unsupervised Key-phrase Extraction using Noun Phrases", *International Journal of Computer Applications*, 162(1), 2017.
- [25] Gadag, Ashwini I., and B. M. Sagar. "N-gram based paraphrase generator from large text document", In *Computation System and Information Technology for Sustainable Solutions (CSITSS)*, *International Conference on*, pp. 91-94. IEEE, 2016.
- [26] Shirakawa, Masumi, Takahiro Hara, and Shojiro Nishio. "N-gram idf: A global term weighting scheme based on information distance", In *Proceedings of the 24th International Conference on World Wide Web*, pp. 960-970. International World Wide Web Conferences Steering Committee, 2015.
- [27] Chatterjee, Niladri, and Neha Kaushik. "RENT: Regular Expression and NLP-Based Term Extraction Scheme for Agricultural Domain", In *Proceedings of the International Conference on Data Engineering and Communication Technology*, pp. 511-522. Springer Singapore, 2017.
- [28] Nesi, Paolo, Gianni Pantaleo, and Gianmarco Sanesi. "A Distributed Framework for NLP-Based Keyword and Keyphrase Extraction From Web Pages and Documents", In *DMS*, pp. 155-161. 2015.
- [29] Onan, Aytuğ, Serdar Korukoğlu, and Hasan Bulut. "Ensemble of keyword extraction methods and classifiers in text classification", *Expert Systems with Applications* 57 pp. 232-247, 2016.
- [30] Habibi, M. and Popescu-Belis, A.. "Keyword extraction and clustering for document recommendation in conversations", *IEEE/ACM Transactions on audio, speech, and language processing*, 23(4), pp.746-759, 2015.
- [31] Rohini P. Kamdi, Avinash J. Agrawal, "Keywords based

Closed Domain Question Answering System for Indian Penal Code Sections and Indian Amendment Laws", *I.J. Intelligent Systems and Applications (IJISA)*, vol.7, no.12, pp.57-67, 2015. DOI: 10.5815/ijisa.2015.12.06

Authors' Profiles



Imtiaz Hossain received the B.Sc. in Information and Communication Technology, Mawlana Bhashani Science and Technology University, Bangladesh, in the year 2013.

He is currently working as Software Engineer in Dohatec New Media an independent software vendor (ISV) and technology firm developing solutions for enterprise clients in North America, Asia and Europe. His research interests include Artificial Neural Network, Data Mining and Machine learning etc.



Asraf Uddin Ahmed received his B.Sc. in Information and Communication Technology, Mawlana Bhashani Science and Technology University, Bangladesh, in the year 2013. He is currently working as Full Stack Developer at Oversea Security and Oversea Digital Media.. His research interests include Artificial Neural

Network, Data Mining, knowledge Engineering etc.



Md. Manowarul Islam received the B.Sc. and M.S. degree in Computer Science and Engineering from the University of Dhaka, Bangladesh, in the year 2008 and 2010, respectively. Currently, he is serving as Assistant Professor in the Department of Computer Science and Engineering, Jagannath University, Bangladesh. He was a former faculty member of Dept. of Information and Communication Technology, Mawlana Bhashani Science and Technology University as a lecturer. He is currently pursuing the Ph.D. degree at the Graduate School of Natural Science and Technology, Okayama University, Japan.

His research interests include the area of modeling, analysis, and optimization of protocols, Data mining, Machine learning, Artificial Neural Network, Wireless Sensor networks etc. He is the corresponding author of this research paper.



Md. Selim Al Mamun received his B.Sc. (Hons) and M.S. degrees in Computer Science and Engineering from the University of Dhaka, Bangladesh in 2007 and 2009 respectively. He joined as a faculty member at the Department of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh in 2011. He is currently pursuing the Ph.D. degree at the Graduate School of Natural Science and Technology, Okayama University, Japan. His research interests include computer networks, optimization algorithms, network security and quantum computing, Big Data, Data Mining.



Md. Ashraf Uddin received the B.Sc. and M.S. degree in Computer Science and Engineering from the University of Dhaka, Bangladesh, in the year 2008 and 2010, respectively. Currently, he is serving as Assistant Professor in the Department of Computer Science and Engineering,

Jagannath University, Bangladesh. He was former faculty member of Dept. of Computer Science and Engineering, Mawlana Bhashani Science and Technology University as a lecturer. He is currently pursuing the Ph.D. degree at the faculty of Information Technology, Federation University, Australia. His research interests include the area of cybersecurity and modeling, analysis, and optimization of protocols, Artificial Neural Network, Big Data, sensor network etc.

How to cite this paper: Imtiaz Hossain Emu, Asraf Uddin Ahmed, Manowarul Islam, Selim Al Mamun, Ashraf Uddin, "An Efficient Approach for Keyphrase Extraction from English Document", *International Journal of Intelligent Systems and Applications(IJISA)*, Vol.9, No.12, pp.59-66 2017. DOI: 10.5815/ijisa.2017.12.06