

# Graph-Based Keyword Extraction

Omar Alqaryouti, Hassan Khwileh, Tarek Farouk, Ahmed Nabhan  
and Khaled Shaalan

**Abstract** Keyword extraction has gained increasing interest in the era of information explosion. The use of keyword extraction in documents context categorization, indexing and classification has led to the emphasis on graph-based keyword extraction. This research attempts to examine the impact of several factors on the result of using graph-based keyword extraction approach on a scientific dataset. This study applies a new model that processes the Medline scientific abstracts, produces graphs and extracts 3-graphlets and 4-graphlets from those graphs. The focus of the experiment is to come up with a dataset that consists of the keywords and their occurrences in the proposed graphlets patterns for each abstract with its class. Then, apply a supervised Naïve Bayes classifier in order to assign a probability to each word, whether or not it is a keyword, and finally evaluate the performance of the graph-based keyword extraction approach. The model achieved significant results compared to the Term Frequency/Inverse Document Frequency (*TF/IDF*) baseline standard. The experimental results proved the capability of using graphs and graphlet patterns in keyword extraction tasks.

---

O. Alqaryouti (✉) · H. Khwileh · T. Farouk · K. Shaalan  
Faculty of Engineering and IT, The British University in Dubai, Dubai, UAE  
e-mail: omar.alqaryouti@gmail.com

H. Khwileh  
e-mail: hassan.khwileh@gmail.com

T. Farouk  
e-mail: tafarouk@me.com

A. Nabhan  
Faculty of Computers and Information, Fayoum University, Fayoum, Egypt  
e-mail: ahmed.nabhan@gmail.com

A. Nabhan  
Member Technology, Sears Holdings, Hoffman Estates, USA

K. Shaalan  
School of Informatics, University of Edinburgh, Edinburgh, UK  
e-mail: khaled.shaalan@buid.ac.ae

**Keywords** Graph-based representation · Graph-based methods  
Graph patterns extraction · Keyword extraction · Machine learning  
Supervised methods

## 1 Introduction

In the last century, graph theory has gained growing and extensive traction in the explosion of computer networks and internet as a well-studied science in the mathematical field. Graph-based representation of text documents allows powerful and comprehensive methods and algorithms such as random walks as well as frequent subgraph mining. This representation facilitates capturing corresponding features for various Natural Language Processing (NLP) applications. Additionally, graph-based ranking methods were proposed to assist in evaluating the importance of a word in a text document with relevance to its adjacent words [8].

In NLP domain, keyword extraction is considered among the most essential key aspects when it comes to text processing. Readers may take the advantage of keywords as it can help them in deciding whether or not to read a document. As for the website developers, they can use keywords in grouping and categorising the website content and materials by its topics.

As a matter of fact, keyword extraction is said to be an effective method applied to many NLP applications. Through extracting main keywords, one may easily select the relevant document to use for learning the relation among the documents. A study by Gutwin et al. [6], the authors described Keyphind; which basically represents key-phrases and keywords from documents; as the essential building block for Information Retrieval (IR) systems. Likewise, Matsuo and Ishizuka [7] pointed out the significance of keyword extraction techniques for various NLP applications, such as document retrieval, Web page retrieval, document clustering, summarization and text mining. Extracting the proper keywords can assist in easily finding out the documents to read as well as learning how documents are related to each other.

Beliga et al. [2] clarified the way in which the keywords are being assigned through terms of controlled vocabulary or predefined classification. Keywords help in describing the main aspects and concepts discussed in a given context. Keyword assignment in simple terms is the process of identifying few words, phrases and terms that can represent a document. There are various approaches of keyword and key-phrases assignment. Authors and Subject Matter Experts (SME's) manually assign keywords and key-phrases to text documents. Though, the approach remains expensive as compared to other options. This approach is a monotonous and time consuming task apart from being expensive. This could be considered the reason behind the importance of automating the keyword extraction process.

In the literature, several approaches were anticipated by researchers for the keyword extraction process. Beliga et al. [2] proposed three methods for keyword extraction; supervised, unsupervised and semi-supervised. The supervised keyword extraction requires a training set and the use of Machine Learning techniques such

as Naïve Bayes and Support Vector Machine. It is domain dependent to the extent that when the domain changes the model will need to be retrained. On the other end of the spectrum, there is the unsupervised methods. At that end there are many statistical-based methods that use frequency based measures, such as *TF/IDF*. These methods are not tied to specific language and does not require training dataset. However, they fare poorly with professional text like health and medical domain; for example, PubMed where a keyword representing medical term might appear rarely.

The following sections will review the work that has been done in the area of keyword extraction in the Related Work Section. Followed by Research Methodology which discusses the concepts of keyword extraction and the proposed model for our research. Based on the approach applied, the experiment will show the essence of proposed approach and its applicability. Then, the Discussion Section reviews the challenges and drawbacks that came across the research project and suggestions for enhancements. Finally, a summary of the impact of using the model for keyword extraction, limitations and future work are discussed in the Conclusion and Future Prospects section.

## 2 Related Work

In this research project, graph-based supervised methods are being examined and experimented for extracting keywords from text documents. According to the known techniques, keywords get assigned from a list of words that are controlled by authors and librarians. The process of extracting keywords attempts to identify the words from the context that are essential and representative of that particular document. Graph-based techniques in most aspects help in exploiting graph structural features to achieve that objective.

Page et al. [11] presented through their research work PageRank, what is considered a graph-based scoring algorithm. The researchers approach uses random-walks algorithms to score a webpage according to its significance that is driven from its interlinks to each other. Likewise, Mihalcea and Tarau [8] established an adaptation of a similar algorithm in the keyword extraction task. The basis of this adaptation is the central aspect of the natural languages words in a certain text. It is also a key element of narrative connection between each other in the same concept of the links between webpages. The relations in NLP are rich and complicated. The Words in NLP consist of relationships, such as phonological, lexical, morphological, syntactical, and semantical. Graph-based methods in most cases are an outstanding choice towards relation representation. Furthermore, it has the capacity to enrich the relationships by using edges with weights, direction, and other elements. The rich illustration can then get examined and learned via Graph Theory. Apart from that, it has the potential of exploiting throughout the Machine Learning techniques.

In a study developed by Erkan and Radev [5], graph random-walks were used in text summarization process. The sentences were assigned to nodes in the graph while the edges were used to represent the cosine similarity for the connected end nodes.

On the other hand, Rose et al. [14] demonstrated a method that divides the abstract based on keywords and key-phrases candidates through stop words and phrase delimiters. The graph is built from nodes and edges where nodes represent words and edges represents the connection between words. The degree of the vertex and frequency of every word are evaluated, i.e. the edges count linked to the vertex gets calculated. The result of every word is illustrated through the ratio where the word frequency gets divided by its degree. A score gets allocated to the key phrases through the summation of the scores of its words. Then, the key phrases get sorted in descending order according to the scores they achieved. Finally, the main key phrases are derived from the topmost scores. The researchers showed that Rapid Automatic Keyword Extraction (RAKE) does not require Part of Speech (POS) tagging. RAKE produced a single aspect algorithm that creates a low cost as well as a high performance and fast algorithm. Thus, the outcome of RAKE algorithm achieved similar results compared to TextRank.

Palshikar [12] proposed a graph-theoretical concept to identify the significance of a word in a given text. The text was represented as undirected graph with words as vertices and the linkage among the adjacent words as edges labelled by its dissimilarity measure. The researcher used a hybrid approach by adopting various algorithms that use eccentricity, centrality and proximity measures to extract keywords. The word (vertex) centrality in the graph is an indicator to its significance as candidate keyword.

Nabhan and Shaalan [9] presented a method for keyword identification through the use of text graphlet patterns. The proposed experimental methodology showed the competence of the graphlet patterns in keyword identification. The authors confirmed the importance of the set technique stances from a suitable data demonstration that increases the context of texts to span various sentences. In that aspect, it allows the attainment of essential topological features of non-local graph. They defined the graphlets as the small efficiently extracted as well as scored sub-graph patterns. These graphlets show the statistical reliance between the graphlet patterns as well as the words that are labelled as keywords.

## 3 Research Methodology

### 3.1 Overview

Ruohonen [15] stated that the graph consists of linked edges and vertices. Thus, a graph is constructed by a set of pairs  $(V,E)$  where  $V$  signifies the set of vertices and  $E$  signifies the set of edges that demonstrates the linkage between two vertices. The degree of the vertex relates to the total count of edges linked to the vertex; where, the vertex is recognized as an end vertex. There are two types of graphs: directed and undirected graphs. Directed graphs have vertices linked through edges

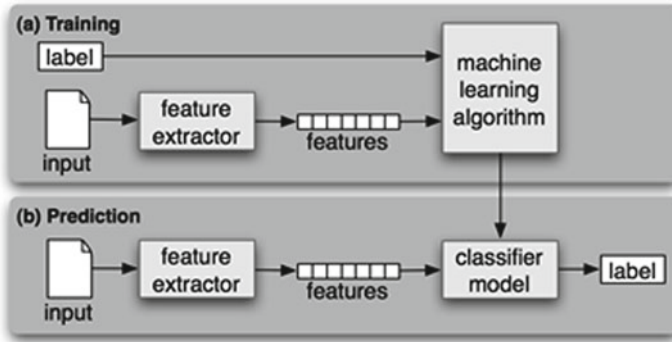


Fig. 1 Supervised classification [3]

with direction. In contrast, the undirected edges do not have the direction between pairs of vertices.

Pržulj [13] defined graphlets as small subgraphs that are linked and non-isomorphic for a large network that allows the capture of local graph or the network topology. A study by DePiero and Krout [4] showed the graph isomorphism and automorphism. These authors claimed that for given graphs  $G$  and  $H$  with  $h_k$  and  $g_i$  nodes correspondingly are isomorphic given the existing mapping of:

$$h_k = m(g_i)$$

This mapping maintains all nodes adjacency. Using this concept, the exact application node and the edge must have consistency with this mapping. The condition for subgraph isomorphism is that an isomorphism between the graph ( $G$ ) and the subgraph ( $H$ ) should exist. The isomorphism can be verified by the rearrangement of the nodes and edges and then the node-to-node mapping using the adjacency matrix.

As illustrated in Fig. 1, Bird et al. [3] stated that supervised Machine Learning methods adopts the use of training and prediction (testing) data to conclude a model that maps the input features with the anticipated result. Thus, this model is expected to correctly predict the anticipated results based on the given features of new data.

### 3.2 The Proposed Methodology

The proposed methodology (as shown in Fig. 2) is meant to produce keywords using graphs through supervised learning methods. The work on this experiment started by selecting and acquiring a large dataset of abstracts and their associated, manually, predefined keywords. Then, an initial pre-processing took place. Afterwards, graphs were built and sub-graphs were identified. These sub-graphs were used to build a feature vector for each word in the abstract. A Naïve Bayes classifier was then applied on the feature vectors dataset. K-fold cross-validation was used to get precision and

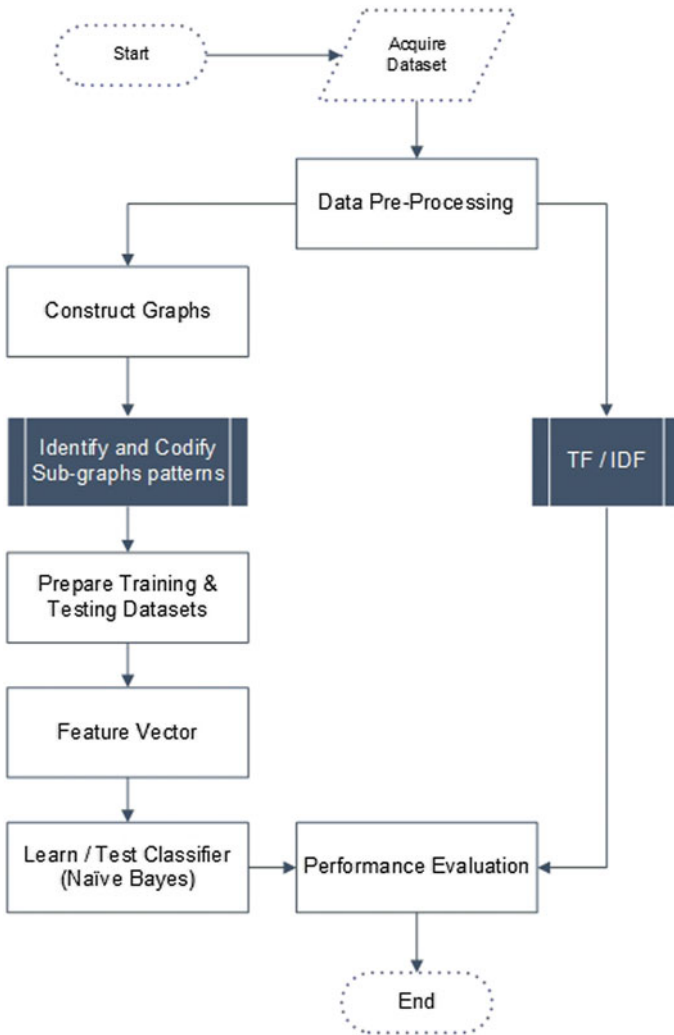


Fig. 2 Research methodology

accuracy averages. The results were then compared to a baseline. The baseline was generated by applying *TF/IDF* on the same dataset.

The process starts by building a dataset from a database of abstracts and their associated preassigned keywords. These abstracts are then pre-processed by taking-out stop words, non-nouns, non-adjectives and words with length less than five characters. Then, abstract graphs are constructed. Each word was represented by a node. The co-occurrence of two words (i.e. adjacent words) was used to connect their respective nodes with an edge. Next, the subgraphs are processed based on the

abstract words to identify and codify the sub-graphs patterns. In this step, a novel idea is proposed which is used to identify subgraphs possible patterns using Depth First Search (DFS) by navigating throughout all nodes and extracting all possible 3-graphlet and 4-graphlet patterns. In general, there are three possible 3-graphlet patterns and eleven possible 4-graphlet patterns. A three-digit-code is used to represent the fourteen patterns as follows:

- The first digit is the degree of the word vertex (most left node as shown in Table 1).

$$d(v_i)$$

where,  $v_1$  is the first vertex.

- The second digit is the sum of degrees of all vertices directly connected to first vertex.

$$\sum_{i=1}^k d(v_i)$$

where,  $v_i$  represents the directly connected vertices to the first vertex and  $k$  the number of vertices that are directly connected to the first vertex.

- The third digit is the sum of degrees of vertices which are connected to the original word through another word only.

$$\sum_{j=1}^q d(v_j)$$





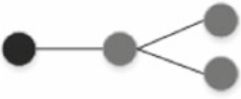
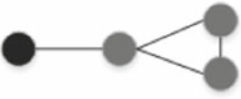




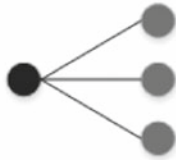
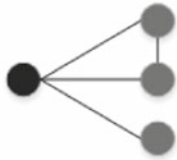
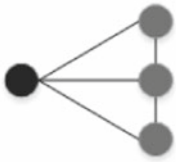
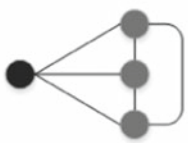
where,  $v_j$  represents the vertices that are indirectly connected to the first vertex and  $q$  is the number of vertices that are indirectly connected to the first vertex.

### 3.3 Dataset

The dataset constitutes of randomly selected abstracts from a scientific dataset that has been obtained from Medline database. PubMed Central is a library of open digital text repository that archives abstracts and references on Medline database and medicine topics. in June 1997 PubMed was published publicly as an open free library. It has more than 26 million records and increasingly about half million records are added yearly. The daily usage of the PubMed is over 3 million searches per day, considered as the world's largest medical library [10].

The same source was also used by Nabhan and Shaalan [9] to process around 10,000 scientific abstracts. Initially, the extracted corpus constituted of around 205,000 abstracts from Medline.

**Table 1** The fourteen possible graphlets patterns and its assigned codes. The left most node is the word vertex that is being evaluated

3-Graphlets		
		
SG121	SG220	SG240
4-Graphlets		
		
SG122	SG132	SG134
		
SG231	SG251	SG242
		
SG262	SG330	SG350
		
SG370	SG390	



### 3.4 *Data Processing*

Each abstract in the PubMed library is accompanied by a keyword list that is provided by either authors or librarians. These pre-assigned keywords were used to train and test the Naïve Bayes classifiers. The process starts by filtering-out abstracts that has less than 3 keywords. Also, a keyword was only considered if it was part of the abstract. Afterwards, the abstract was tokenized using NLTK library similar to Bird et al. [3]. Tokenization was followed by a POS tagging process with a filter on the tagged words to exclude all non-nouns and non-adjectives and words with length less than five characters.

In the same way to the TextRank algorithm proposed by Mihalcea and Tarau [8] to mandatory define relationship in which was utilized to have one relationship in this experimental study. Thus, a relationship was constructed to discover the co-occurrence of the words within a predefined range of two adjacent words that identifies the sequences of word pairs that create the un-weighted edges in the abstract graph. An undirected graph will be constructed using the processed words and edges (words pairs) ignoring the pairs order and direction as long as the words are adjacent.

### 3.5 *Learn Classifier and TF/IDF*

The proposed model intended to process the produced graphs for the whole corpus as well as identify and codify the sub-graphs possible patterns. An algorithm has been implemented using Python to identify sub-graphs patterns using Depth First Search (DFS) by navigating through all nodes and extracting all possible 3-graphlet and 4-graphlet patterns (see Table 1). The solution uses NLTK, NetworkX, Biopython API's and Rapid Miner. For each word vertex, a mechanism was built for reiteration through the graph to get all possible 3-graphlets and 4-graphlets that the word vertex is part of. This ends up with a new dataset that contains the word, all possible 3-graphlet and 4-graphlets patterns with the count of the number of occurrences for the word in this particular pattern. Moreover, the dataset includes a classification of whether the word was identified as keyword or non-keyword according to the abstract Other Terms (OT's). This training set was used as a data source for the Naïve Bayes classifier. The classifier will assign a probability to each word depending on whether or not it is a keyword. RapidMiner tool was used to implement the supervised classification model of Naïve Bayes (see Figs. 3, 4 and 5).

Gaussian Naïve Bayes has been used in the results evaluation. The continuous values that involve the graphlet patterns counts for each word and associated with a class for the word being a keyword or not being a keyword are distributed according to Gaussian Naïve Bayes. For instance, consider a training dataset that includes continuous attribute. The first step is to cluster the data according to the class and after that compute the mean and variance of the attribute in each class. Assume, for

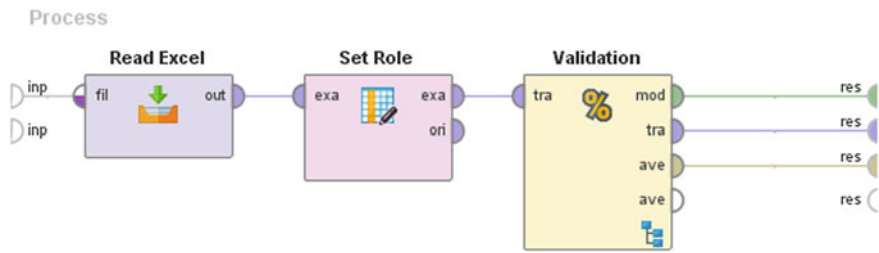


Fig. 3 RapidMiner main process

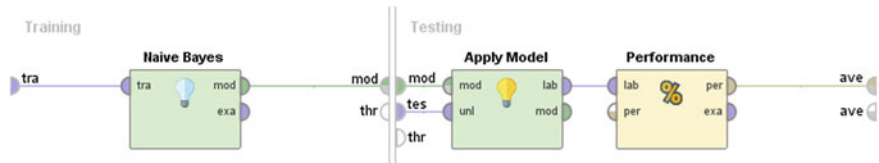


Fig. 4 Naive Bayes subprocess (Validation)

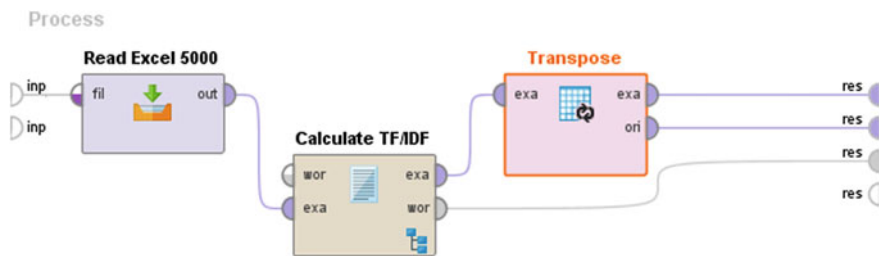


Fig. 5 RapidMiner process to calculate the *TF/IDF* scores

a word from the test dataset, that  $v$  is the count of graphlet  $x$ . Then, the probability distribution of  $v$  given a class  $c$  can be computed according to the following formula:

$$p(x = v|c) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

where,  $\sigma_c^2$  is the variance of all counts of the graphlet  $x$  in the training dataset that is linked to class  $c$ , and  $\mu_c$  is the mean of all values in  $x$  linked in class  $c$ .

In this research study, *TF/IDF* has been used as a baseline standard. Aizawa [1] claimed that in recent information retrieval systems, *TF/IDF* is among the most regular and widely used term weighting techniques. Apart from *TF/IDF* fame, it has been considered as an empirical approach which is a frequency-based approach that considers the words and document frequencies.

The *TF/IDF* is known with its well reasonable performance and efficiency. It consists of three elements: *TF*, *IDF* and *TF/IDF*. The *TF* for a particular word is the number of occurrences for this word  $W_i$  in certain document. Whereas, the *IDF* value can be calculated according to the following formula:

$$idf = \log\left(\frac{N}{N_i}\right)$$

where,

- $N$  represents the entire number of documents in the dataset,
- $N_i$  represents the total number of documents that include the word  $W_i$

Finally, the *TF/IDF* score for a word in a document which is the final product of both *TF* and *IDF* that can be determined using the following formula:

$$tfidf = tf \times idf$$

In order to get higher *TF/IDF* score, there shall be a higher *TF* score for the word in a particular document with a lower document frequency of the same word in the entire dataset.

The *TF/IDF* approach has been used to identify the keywords for each document. As for this research, it has been assumed to identify the top six words as the selection criteria for the candidate keywords. As for abstracts, the authors are usually allowed to define six keywords for their publications. In particular, once the *TF/IDF* scores were calculated for all words in the entire documents collection, the candidate keywords were nominated according to the highest *TF/IDF* scores. A process in RapidMiner has been designed to calculate the *TF/IDF* scores (see Fig. 5). The output of the process has been followed by an automated process in Microsoft Excel 2016 to sort the words to get the top five candidate keywords and then calculate the precision and recall percentages according to the OT values.

### 3.6 Performance Evaluation

The keywords outcome of RapidMiner process for both learn classifier and *TF/IDF* baseline standard has been used for performance evaluation. These evaluation results for both the proposed model and baseline standard were compared to measure the performance of the proposed model.

The identified keywords have been compared with the annotated keywords that are assigned to each abstract (OT's) to generate the confusion matrix and calculate the Precision ( $P$ ) and Recall ( $R$ ) accuracy measures. The precision percentage indicates the amount of identified keywords that were relevant and its value was calculated according to the following formula:

$$P = \frac{TP}{TP + FP}$$

where,

- *TP* is the True Positive count which illustrates the keywords that were identified using the proposed supervised model or the *TF/IDF* algorithm which were part of the keywords that were assigned to the abstract (OT's).
- *FP* is the False Positive count which is also known as Type I error that illustrates the keywords that were identified using the proposed supervised model or the *TF/IDF* algorithm which were not part of the keywords that were assigned to the abstract (OT's).

Additionally, the Recall percentage value was calculated according to the following formula:

$$R = \frac{TP}{TP + FN}$$

where, *FN* is the False Negative count which is also known as Type II error that illustrates the keywords which were part of the keywords that were assigned to the abstract (OT's) but were not identified using the proposed supervised model or the *TF/IDF* algorithm.

Finally, the F-Measure also known as F-Score represents a combination of Precision and Recall values into a single measurement value. The F-Measure value was computed using the following formula:

$$F - Measure = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)}$$

Table 2 demonstrates the performance evaluation results for both the learned classifier that used the graph-based representation and graphlets frequent pattern identification in addition to the *TF/IDF* baseline standard. The evaluation outcome showed significant results for the proposed model compared to the *TF/IDF*. The results of the proposed model have achieved 76.32% for Precision, 62.88% for Recall and 68.95% for F-Measure. While *TF/IDF* has achieved 64.40% for Precision, 56.48% for Recall and 60.18% for F-Measure.

This section discussed in details the proposed model and how to produce keywords using graphs and graphlet patterns through supervised learning methods. Furthermore, it explained the methodology that was followed and the various work on this experiment throughout the dataset selection, the design and development phase

**Table 2** Performance evaluation results

System	Precision (P) (%)	Recall (R) (%)	F-Measure (%)
TF/IDF baseline	64.40	56.48	60.18
Graph representation	76.32	62.88	68.95

and data preparation and validations. Lastly, it illustrated the performance evaluation results.

## 4 Discussion

This study aims to experiment the impact of using graph-based techniques and sub-graphs patterns identification in keyword extraction. A novel efficient method was introduced for exploring significant patterns in word graphs. The dataset used was initially consisted of around 205,000 abstracts. The dataset was filtered to exclude those abstract that does not contain the OT as part of it. Additionally, the abstracts that does not have OT's assigned to them was also excluded. The processed dataset ended up with around 25,000 abstracts after the original dataset was filtered.

The proposed model assumes that the keywords should be part of the documents and does not consider the keywords that are not part of the documents. A better algorithm should not restrict its word space to the given abstract or document but rather should leverage the whole corpus or at least the whole space of keywords of a given corpus. The challenge would be finding strong predictors from a given text that point at the most probable keywords from a pool of keywords from the corpus. These predictors can take the shape of labelled graphlets. Another way could be through embedding the keyword in a graph that connects it to selected words from the abstract. This will give us a database of graphs for each keyword. New abstract can be tested against this database assigning a proximity weight for each keyword in the pool. Then, the candidate keywords can be selected from the most top ones after sorting keywords based on the proximity weight.

## 5 Conclusion and Future Prospects

Keywords have been proved to be important for document retrieval, text summarization, retrieval of webpages and text mining. Moreover, the keywords help in attracting readers to easily select the relevant topics and documents to read. As stated by Matsuo and Ishizuka [7], the keyword extraction techniques have significant impact on various NLP tasks. The study has experimentally illustrated the significance of graph-based text representation and graphlets patterns approaches in keyword extraction. The results have showed a capability of using graph text representation in extracting keywords compared to the state-of-the-art *TF/IDF*.

There are several areas that can attract researchers for further elaboration in extending the model and the experiment framework with other options that may improve and enhance its outcome. These options may spread to emphasize weighted graphs based on the number of co-occurrences allowing 3-gram and 4-gram, proper treatment of in-line equation and special characters, include edge information like weights and labels [9], measure the impact when extending to use 5-graphlet pat-

terns, cover multi-word keywords/phrases [9], the use of graphs in Text Categorisation [9] and adjoin the extracted keywords that have “and” and “of” in between them which should improve the readability of the key phrases [14]. Also, exploring techniques to mine a corpus for possible keywords for a given abstract rather than restricting the algorithm to words appeared in the given abstract.

Though this experiment used abstracts to automatically extract keywords, it can be equally applied to full-text articles. How effective and efficient is this method on full-text is another good point of future research.

## References

1. Aizawa, A.: An information-theoretic perspective of tf-idf measures. *Inf. Process. Manag.* **39**(1), 45–65 (2003)
2. Beliga, S., Meštrović, A., Martinčić-Ipšić, S.: An overview of graph-based keyword extraction methods and approaches. *J. Inf. Organ. Sci.* **39**(1), 1–20 (2015)
3. Bird, S., Klein, E., Loper, E.: *Natural language processing with Python*. O’Reilly Media, Inc. (2009)
4. DePiero, F., Krout, D.: An algorithm using length-r paths to approximate subgraph isomorphism. *Pattern Recogn. Lett.* **24**(1), 33–46 (2003)
5. Ergan, G., Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.* **22**, 457–479 (2004)
6. Gutwin, C., Paynter, G., Witten, I., Nevill-Manning, C., Frank, E.: Improving browsing in digital libraries with keyphrase indexes. *Decis. Support Syst.* **27**(1), 81–104 (1999)
7. Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. *Int. J. Artif. Intell. Tools* **13**(01), 157–169 (2004)
8. Mihalcea, R., Tarau, P.: *TextRank: bringing order into texts*. *Assoc. Comput. Linguist.* (2004)
9. Nabhan, A.R., Shaalan, K.: Keyword identification using text graphlet patterns. In: *International Conference on Applications of Natural Language to Information Systems*, pp. 152–161. Springer (2016)
10. Ncbi.nlm.nih.gov. Home-pubmed-ncbi. <http://www.ncbi.nlm.nih.gov/pubmed>, August (2016)
11. Page, L., Brin, S., Motwani, R., Winograd, T.: *The Pagerank Citation Ranking: Bringing Order to the Web* (1999)
12. Palshikar, G.K.: Keyword extraction from a single document using centrality measures. In: *International Conference on Pattern Recognition and Machine Intelligence*, pp. 503–510. Springer (2007)
13. Pržulj, N.: Biological network comparison using graphlet degree distribution. *Bioinformatics* **23**(2), e177–e183 (2007)
14. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents. *Text Mining*, pp. 1–20 (2010)
15. Ruohonen, K.: *Graph theory, graafiteoria lecture notes*, tut (2013)