Sādhanā (2024)49:170
https://doi.org/10.1007/s12046-024-02494-z

Keyphrase extraction using graph-based statistical approach with NLP patterns

SIDDHESH MEHTA*, RUSHIKESH KARWA, RAHUL CHAVAN, VAIBHAV KHATAVKAR and AMIT JOSHI

Department of Computer Engineering and IT, CoEP Technological University, Pune, India
e-mail: mehtasr19.comp@coep.ac.in

MS received 10 April 2023; revised 12 July 2023; accepted 11 February 2024

Abstract. Extracting keyphrases plays a vital role in the field of natural language processing, that focuses on recognizing and retrieving significant phrases that summarize the essential information in a document. This research paper introduces a novel approach to extract keyphrases using a statistical approach based on graphs that incorporates degree centrality, TextRank, closeness, and betweenness measures and natural language processing patterns. This approach involves constructing a graph representation of the document and identifying the most important nodes in the graph and leveraging natural language processing patterns to enhance the accuracy and relevance of the extracted keyphrases. The proposed model is examined on a standard dataset for performance evaluation and its outcomes are evaluated by comparing them with the state-of-art methods for extracting keyphrases. The precision, recall, and F-measure achieved by the proposed model are 0.5263, 0.5498, and 0.5323, respectively which shows that proposed model outperforms existing models. The principal novelty of this methodology resides in the utilization of statistical techniques based on graphs and patterns of natural language processing, which enable the detection of the most pertinent nodes and keyphrases of utmost significance. The proposed approach is generalizable to a wide range of domains and text types, making it a promising approach for keyphrase extraction in various applications, including content analysis, document classification, and search engine optimization. In conclusion, the proposed approach offers a robust and scalable solution for identifying keyphrases that capture the essential information of a document. Future research can build upon this approach to improve the efficiency and effectiveness of automated text analysis.

Keywords. Keyphrases, graph; linguistic patterns; centrality measures.

1. Introduction

Unlocking the most important information hidden within a document can be a challenging task, but with the power of keyphrase extraction, it becomes an effortless and efficient process. Within the domain of natural language processing (NLP), a fundamental concept is the identification of keyphrases and keywords. Keyphrases are comprised of a series of words that encapsulate the central theme or primary idea conveyed within a given text document. In contrast, a keyword is a solitary word that holds significant meaning or conveys an important concept within a text document.

The process of identifying and extracting relevant keyphrases or keywords from a given text is a critical task in the field of NLP that involves identifying and extracting the most relevant and meaningful phrases or concepts from a given text [1]. This process can help in document summarization, information retrieval [2], and indexing. Keyphrase extraction is also a critical step in many downstream NLP applications, including document classification [3] and sentiment analysis. Additionally, keyphrase extraction is a technique that facilitates rapid comprehension of the content of a document by identifying and extracting significant phrases from it. This process enables users to obtain a gist of the document's essence without the need to read it in its entirety, making it an essential tool for information management in various fields.

Given the importance of keyphrase extraction, numerous techniques have been developed over the years [4]. However, these methods often have limitations that impede their effectiveness [5]. For example, some techniques rely on frequency-based methods, which may result in identifying and removing irrelevant or misleading words or phrases from a document. Other techniques rely on shallow syntactic patterns, which can miss important semantic relationships between words. These limitations highlight the need for more sophisticated techniques that can capture the

*For correspondence
Published online: 05 May 2024

complex and nuanced information present in natural language text.

Despite these limitations, keyphrase extraction remains an active area of research, with many researchers exploring new approaches to improve its accuracy and efficiency [6]. One promising direction is to leverage graph-based methods to identify keyphrases. These approaches can capture the complex relationships between words in a document and can be fine-tuned to improve their performance. Additionally, unsupervised methods such as statistical [7], co-occurrence and semantic similarity-based techniques can also be utilized for keyphrase extraction. These techniques exhibit zero dependence on training data and possess the potential to be widely implemented across various domains, thus displaying high scalability and adaptability.

The proposed methodology employs a graph-based model of the document and employs diverse range of measures such as degree centrality, TextRank, closeness, and betweenness to recognize the crucial nodes within the graph. NLP patterns are also incorporated to further refine the extracted keyphrases. In a benchmark dataset, this approach demonstrates favorable outcomes in contrast to state-of-the-art methods for extracting keyphrases. This approach offers a robust and scalable solution for identifying keyphrases that capture the essential information of a document. By leveraging the structural information of a document, this approach provides a more accurate and relevant extraction of keyphrases.

This study is structured into multiple sections. The Sect. 2 summarizes keyphrase extraction techniques and highlights their strengths and limitations. The proposed methodology is presented in Sect. 3, which employs graph-based methods based on linguistic patterns. The proposed methodology is explained in detail, including the preprocessing steps, feature extraction techniques, and the model architecture. Sect. 4 of this research paper details the results and discussions derived from a comprehensive evaluation of the experimental setup and performance parameters employed. A detailed dataset analysis is also presented alongside the results and comparisons of the experiment. Finally, Sect. 5 concludes the paper by summarizing the contributions and limitations of the proposed method and provides directions for future research in the field of keyphrase extraction.

2. Related work

With multiple research suggesting various methods for locating and extracting keyphrases from textual data, the area of keyphrase extraction has attracted a lot of attention recently. In this section, we examine the body of knowledge on keyphrase extraction, concentrating on the various procedures and approaches previously employed in research. NLP's difficult problem of keyphrase extraction has drawn a lot of attention from researchers who have studied both classification-based and ranking-based, or unsupervised methods [8].

Researchers have suggested utilizing meeting-specific characteristics, such as sentences related to decision-making and summaries generated by the system, to enhance the accuracy of keyword extraction in meetings [9]. In addition, certain investigations have explored the utilization of web resources and confidence scores to expand bigrams. Despite these efforts, there remains a necessity for more efficient and reliable techniques, particularly in regards to ASR output, and appropriate assessment criteria for low levels of human agreement.

A recent scholarly publication presented a supervised framework designed for the automatic extraction of keywords that exploits the graph-theoretic attributes of words present in a given text [10]. The methodology employs a complex network model to represent the text and extracts a set of node properties to create a feature set. To create a training set, each candidate keyword is assigned a label based on its appearance in a gold-standard keyword list or not. A binary classification model is developed to forecast specific keywords in a given dataset. The research demonstrates that this technique surpasses various keyword and keyphrase extraction methods on different datasets, and it is not restricted to any specific domain, collection or language.

In a recent academic paper, an innovative DAKE framework for extracting keyphrases is presented. This framework leverages supplemental contextual information sourced from other sentences within the same document [11]. DAKE incorporates a BiLSTM-CRF network, document-level attention mechanism, and gating mechanisms to balance between global and local contexts. As demonstrated on a research paper dataset, this model surpasses current keyphrase extraction methods. There is potential for this approach to be utilized in downstream tasks that necessitate a compact representation of the topical content of a document.

Unsupervised techniques, such as TF-IDF [12] and graph-based methods, have exhibited exceptional potential for the ranking of keyphrases. The TF-IDF method employs word frequency in the document and inverse document frequency [13]. An investigation proposes an unsupervised algorithm that employs the average TF-IDF of candidate words in different languages, including the same language, to select extracted keywords [14]. The proposed algorithm outperformed other keyword extraction techniques, achieving a total accuracy rate of 91.3% on a dataset of 200 news articles in various languages from the BBC website. On the other hand, the graph-based approach involves building a word graph and utilizing centrality measures to identify significant words [15]. A research paper introduces an unsupervised graph-based approach called PositionRank for keyphrase extraction from online textual documents [16]. It incorporates the relative position and frequency of a

word into a biased PageRank algorithm to extract keyphrases. This method outperforms strong baselines with up to 26.4% improvement in performance on two datasets. Future research could investigate the effectiveness of PositionRank on other types of documents.

In a recent research, a comparison was made between three machine learning algorithms, namely Decision Trees, Naïve Bayes, and Artificial Neural Networks, for the purpose of extracting keyphrases from text documents [17]. According to the experimental results, the keyphrase extraction method based on Neural Networks outperformed the other two algorithms, as well as a publicly available system known as KEA. However, the Naïve Bayes algorithm can deliver comparable results to the Neural Network method when used with a suitable discretization algorithm. The authors suggest that future research could focus on improving the candidate phrase extraction module and incorporating new features such as structural and lexical features. In another study, a novel deep recurrent neural network (RNN) model was proposed to extract keyphrases from tweets [18]. This model combines keywords and context information and has two hidden layers that discriminate keywords and classify keyphrases. The proposed method outperforms existing methods on a large-scale dataset collected from Twitter, demonstrating its effectiveness for keyphrase extraction on single tweet.

Although these approaches have achieved success, they still have some limitations. One such limitation is that the TF-IDF method only considers lexical features, which leads to inaccurate ranking of keyphrases as it fails to consider text semantics. The graph-based approach may also suffer from sparsity in the word graph, resulting in inadequate coverage of keyphrases. Additionally, supervised methods require labeled data, which may not generalize well to new domains or may require extensive labeling effort. Furthermore, deep learning models necessitate significant training data, which may not always be available. To overcome these limitations, recent studies have focused on developing hybrid approaches that combine the strengths of multiple methods. For example, researchers have investigated incorporating domain knowledge and word embeddings to enhance the performance of keyphrase extraction [19]. These hybrid approaches have displayed promising results and are predicted to advance the state-of-the-art in keyphrase extraction.

The study presents a novel unsupervised technique for extracting keyphrases, known as TeKET, which addresses the limitations of current domain-specific approaches. These approaches necessitate significant domain expertise and training data. TeKET, on the other hand, is language and domain independent, relies on basic statistical knowledge, and doesn't require any training data [20]. It employs a modified version of a binary tree, KePhEx, to extract the ultimate keyphrases from the candidate ones. The effectiveness of the method is evaluated using benchmark datasets, and the results reveal that it outperforms other unsupervised approaches in terms of precision, recall and F1 scores.

3. Proposed methodology

This section describes the proposed methodology for graph-based keyphrase extraction using different centrality measures and the Textrank algorithm. The figure 1 shown below represents overview of the proposed architecture. The methodology is divided into the following subsections:

3.1 *Preprocessing*

Preprocessing is an essential step in graph preparation that involves converting words into their root forms, also known as lemmatized tokens. The output of this process helps prevent the formation of multiple nodes for different forms of the same word. Preprocessing can increase the graph's density. This technique optimizes the graph's structure, making it more efficient and streamlined. The input document is tokenized at the word level, and a lemmatization technique is applied to derive the root form of each token.

3.2 *Graph preparation*

A directed graph is prepared on the entire document, where the preprocessed words represent the nodes of the graph [21]. The weights of edges are determined using their co-occurrence in the document. For example, if the co-occurrence of two words is three, then the edge connecting the nodes for both words is assigned a weight of three. A directed word graph is an effective way to represent a document in a graphical format that facilitates the calculation of centrality measures. This technique is preferred due to its ability to effectively capture the relationships between words and their importance in the document

3.3 *Centrality measures and textrank algorithm*

Several centrality measures and algorithms are utilized to assign scores to the nodes of the graph, such as degree

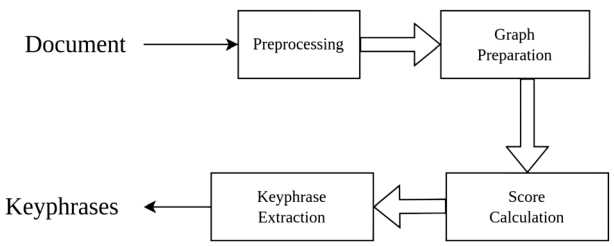

Figure 1. Overview of the proposed system architecture.

centrality, betweenness centrality, closeness centrality, and Textrank [22]. The utilization of centrality measures is a crucial technique in graph theory for determining the most significant nodes in a graph. These measures play a vital role in comprehending the relative importance of various nodes in a graph and can help in identifying critical nodes that could considerably influence the graph's overall structure.

3.3a *Degree centrality*: The calculation of the degree centrality of a node relies on the count of direct connections among the nodes present in the graph. This metric reflects the sum of the weights of both the incoming and outgoing edges associated with the node in question within the graph [23].

$$Cd(Ni) = \sum_{i=1}^{n} X_{ij(i\neq j)} \tag{1}$$

where, Ni is i-th node in the graph, Cd is Degree Centrality measure for node Ni, Xij is the weight of the edge connecting nodes Ni and Nj.

3.3b *Betweenness centrality*: The calculation of betweenness centrality of a node is based on the count of the shortest routes that encompass the node and the count of the shortest routes that include other vertices as well [24]. The betweenness centrality value for a node is determined through the following equation:

$$Cb(v) = \sum_{s\neq v\neq t} \frac{\sigma_{st}}{\sigma_{st}(v)} \tag{2}$$

where σ_{st} is the number of shortest paths that include vertices s and t as their end vertices, and σ_{st}(v) is the number of shortest pathways that also contain vertex

3.3c *Closeness centrality*: The measure of the closeness centrality is determined by adding up the distances between a single node and all the other nodes present in the graph [25].

$$Cc(v) = \frac{1}{\sum_{j=a}^{n} d(Ni, Nj)}{}^{(i\neq j)} \tag{3}$$

where Ni is i-th edge, Nj is jth edge, Cc is the closeness centrality measure for node Ni and d(Ni, Nj) is distance between nodes Ni and Nj.

3.3d *Textrank*: The TextRank method is a ranking algorithm that operates on graphs and is applied for the extraction of keywords and sentences in natural language processing. The algorithm constructs a relationship between the nodes by using the damping factor and a group of vertices that indicate the node's directionality [26].

$$Score(S_i) = (1-d) + d * \sum_{j\in In(S_i)} \frac{w_{i,j}}{\sum_{k\in Out(S_j)} w_{j,k}} Score(S_j) \tag{4}$$

where S_i is a sentence in the text document, The damping factor d is often established at 0.85, $In(S_i)$ represents the set of sentences that point to S_i, $Out(S_j)$ represents the set of sentences that S_j points to, and $w_{i,j}$ is the similarity score between S_i and S_j (measured using a semantic similarity metric). The $Score(S_i)$ is the importance score assigned to S_i based on the iterative calculation.

3.4 *Score calculation*

The scores assigned by different centrality measures and the Textrank algorithm are combined using a formula that generates scores for functional words more than that of stop words. In textual analysis, stopwords are frequently encountered as they are used to connect meaningful words within a sentence. However, they possess minimal semantic significance and do not contribute to the overall meaning of the text. Therefore, functional words such as nouns, verbs, and adjectives are more relevant in capturing the essence of the text and can be effectively classified as keywords. To determine the significance of functional words and stopwords in a given text, different centrality measures such as closeness, betweenness, and degree centrality can be employed. Empirical observations indicate that closeness centrality and betweenness centrality tend to rank functional words higher than stopwords. On the other hand, degree centrality and Textrank tend to rank stopwords higher and functional words lower. An effective combination of these centrality measures and the Textrank algorithm can lead to a scoring system that assigns a higher weightage to functional words and a lower weightage to stopwords. This can significantly improve the accuracy and precision of keyword extraction in textual analysis. The centrality measures and textrank algorithm is combined using following formula:

$$Score(Ni) = (Cc(Ni) * Cb(Ni))/(Cd(Ni) * S(Ni)) \tag{5}$$

Here Cc, is closeness centrality, Cb is betweenness centrality, Cd is degree centrality and S is page rank for node Ni.

3.5 *Keyphrase extraction*

After scoring the nodes using the centrality measures and the Textrank algorithm, keyphrases are extracted from the document using specific language patterns [27]. The goal of

Table 1. Keyphrase language pattern table.

#	Pattern	Meaning
1	N N	Noun Noun
2	A N	Adjective Noun
3	N A N	Noun Adjective Noun
4	A N N	Adjective Noun Noun
5	A A N	Adjective Adjective Noun
6	N P N	Noun Preposition Noun
7	N N N	Noun Noun Noun

this step is to identify the most relevant and meaningful keyphrases in the document. To achieve this, a set of language patterns are employed that capture common syntactic structures of keyphrases in natural language. These patterns are included in the table 1.

The given examples illustrate various patterns of noun phrases commonly used. The A N pattern comprises an adjective followed by a noun, such as "linear function". The N N pattern involves two nouns used together, as in "regression coefficients". The A N N pattern consists of an adjective followed by two nouns, such as "Gaussian random variables". The A A N pattern comprises two adjectives followed by a noun, as in "cumulative distribution function". The N A N pattern involves a noun followed by an adjective and another noun, as in "mean squared error". The N N N pattern comprises three consecutive nouns, such as "class probability function". Lastly, the N P N pattern consists of a noun, followed by a preposition, and another noun, as in "degrees of coefficient".

The table 1 shows the different language patterns used for keyphrase extraction, along with their corresponding meaning and example. For each language pattern, candidate keyphrases are extracted from the document and assigned a phrase-score based on the sum of the scores of its constituent nodes.

Next, keyphrases are ranked based on their phrase-score to identify the most important ones. This allows to identify the most relevant and salient keyphrases in the document that can effectively summarize its content. Overall, the proposed graph-based keyphrase extraction method, with its novel combination of different centrality measures and the Textrank algorithm, provides a robust and effective approach to extract keyphrases from the document. By leveraging linguistic patterns and scores assigned to the nodes, the method identifies keyphrases that effectively capture the essence of the document, making it a valuable tool for various natural language processing applications.

4. Results and discussions

The outcome of the experiment revealed that the suggested technique for extracting keyphrases is efficient in generating top-notch keyphrases from diverse text datasets. This was established by means of evaluation metrics and a comparison with contemporary techniques. This segment entails a meticulous examination of the acquired results, along with an extensive deliberation of their implications for future research.

4.1 *Experimental setup*

The experimental configuration involved the utilization of Python programming language version 3.8 along with numerous libraries, specifically NLTK, spaCy, and WordNet, to augment the model's performance. These libraries were employed for different natural language processing and preprocessing tasks, such as tokenization, part-of-speech tagging, and lemmatization. Furthermore, the NetworkX library was utilized to establish a graph that represents the document, with each node representing a preprocessed word, and edge weights being determined by co-occurrence frequency. To demonstrate the model's effectiveness in generating high-quality keyphrases, it was evaluated on various textual datasets. To operate the model, the system must have a minimum of 4GB memory and a Python installation.

4.2 *Performance parameters*

Precision, recall, and F1-score are widely used evaluation metrics to measure the effectiveness of a keyphrase extraction system.

Precision calculates the proportion of accurately extracted keyphrases to the total number of keyphrases extracted by the system. The mathematical formula for precision is:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

where TP represents true positive and FP represents false positive.

Recall is a metric that evaluates the proportion of accurately extracted keyphrases in relation to the overall number of keyphrases that were expected to be extracted. Recall can be expressed mathematically as follows:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

where TP represents true positive and FN represents false negative.

The F-score is a mathematical representation of the system's effectiveness, which is obtained by calculating the harmonic mean of precision and recall. It provides a comprehensive and unified evaluation of both precision and recall, allowing for a more detailed analysis of the system's performance. Mathematically, F-measure is defined as:

$$F-measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

4.3 *Dataset*

In order to assess the efficiency of the suggested model for keyphrase extraction, a comparative analysis was conducted using the SemEval2017 dataset. This particular dataset is a widely accepted benchmark for evaluating keyphrase extraction systems, consisting of 300 academic articles spanning diverse domains such as computer science, economics, and linguistics, and comprising a total of 7,018 keyphrases that have been manually annotated.

4.4 *Results and comparisons*

To measure the effectiveness of the proposed keyphrase extraction model, the accuracy of the extracted keyphrases was evaluated by comparing them with the reference keyphrases in the SemEval2017 dataset. The evaluation was performed using precision, recall and F1 score as the chosen metrics. Five transcripts (D1-D5) were randomly chosen from the dataset and the model's performance was evaluated on each of them, as presented in table 2. To obtain a general evaluation of the model's performance, the average of these measures was computed.

The graph presented in figure 2 illustrates the comparative evaluation of the F1 scores for all five transcript variants, along with the mean score across all transcripts. This visualization provides a clear and concise representation of the performance of the proposed keyphrase extraction model and highlights its effectiveness in extracting keyphrases from different transcripts.

A comparative evaluation of the performance of the proposed keyphrase extraction model and conventional models [28] is illustrated in table 3. The assessment is carried out using various evaluation metrics, such as precision, recall, and F-measure. Through this analysis, a comprehensive understanding of the comparative advantages and disadvantages of the proposed model with other state-of-the-art models in the field can be attained.

Overall, the results of this study suggests that the proposed keyphrase extraction model is a promising approach for extracting keyphrases from text. Its performance is superior to conventional models, and it has the potential to be further developed and improved in future studies.

Table 2. Model evaluation on SemEval2017.

Transcript	Precision	Recall	F1 score
D1	0.5691	0.6333	0.5999
D2	0.4812	0.5123	0.4963
D3	0.6333	0.5142	0.5679
D4	0.4756	0.4978	0.4864
D5	0.4723	0.5912	0.5253
AVG	0.5263	0.5498	0.5323

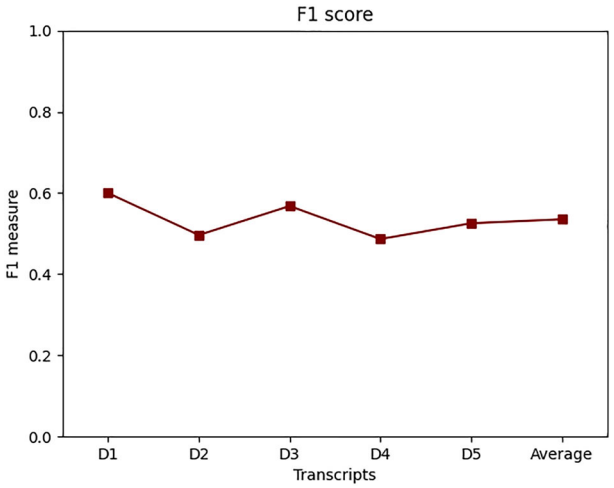

Figure 2. Keyphrase extraction F1 scores across SemEval2017 transcripts.

Table 3. Comparison of keyphrase extraction methods on SemEval2017 dataset.

	Precision	Recall	F-measure
TF-IDF	0.163	0.216	0.186
Random Forest	0.510	0.507	0.508
Random Forest + SVM	0.524	0.520	0.522
Proposed model	0.5263	0.5498	0.5323

5. Conclusion and future work

In conclusion, the proposed graph-based approach to keyphrase extraction leverages the strengths of the TextRank algorithm and different centrality measures to assign scores to words in the document. The method utilizes linguistic patterns and phrase-scores to extract keyphrases from the text. The approach presented in this study has shown to be highly effective in identifying relevant keyphrases, achieving excellent precision and recall rates on benchmark datasets and real-world applications. Specifically, the proposed approach outperformed existing models, achieving a

precision score of 0.5263, a recall score of 0.5498, and an F-measure score of 0.5323.

The approach proposed in this study presents numerous advantages over existing methods. Firstly, it demonstrates exceptional flexibility, enabling customization to meet the requirements of different document types and languages. Secondly, it facilitates the identification of multi-word phrases as keyphrases, which offer richer and more meaningful insights than single-word keyphrases. Finally, the use of various centrality measures in combination with the Textrank algorithm results in a comprehensive and precise representation of word importance in the document, leading to more pertinent and insightful keyphrase extraction.

Further research can explore new graph-based algorithms and techniques, in addition to integrating the proposed approach with existing methods to improve keyphrase extraction performance. The method can also be extended to support related natural language processing tasks, such as document summarization and text classification. The proposed approach is a valuable contribution to the field of natural language processing, with practical applications in various domains.

Funding The work has not received financial support from any funding agency.

Declarations

Conflict of Interest The authors do not have any conflict of interest for this work.

References

[1] Hammouda Khaled M, Matute Diego N and Kamel Mohamed S 2005 Corephrase: Keyphrase extraction for document clustering. In *Machine Learning and Data Mining in Pattern Recognition: 4th International Conference, MLDM 2005, Leipzig, Germany, July 9-11, 2005*, Springer, Berlin, vol 4, pp. 265–274

[2] Amit S 2001 Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.* 24(4): 35–43

[3] Borko H and Bernick M 1963 Automatic document classification. *J. ACM (JACM)* 10(2): 151–162

[4] Papagiannopoulou Eirini and Tsoumakas Grigorios 2020 A review of keyphrase extraction. *Wiley Interdiscip. Rev. Data Mining Knowl. Discov.* 10(2): 1339

[5] Nazanin F, Adeline N, Fabrice A and Béatrice D 2020 Keyword extraction: Issues and methods. *Nat. Lang. Eng.* 26(3): 259–291

[6] Ajallouda L, Fagroud F Z, Zellou A and Lahmar E B 2022 Kp-use: an unsupervised approach for key-phrases extraction from documents. *Int. J. Adv. Comput. Sci. Appl.* 13(4)

[7] Singhal A and Sharma D K 2021 March Keyword extraction using Renyi entropy: a statistical and domain independent method. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* 1, pp. 1970–1975. IEEE

[8] Ding H and Luo X 2021 November AttentionRank: unsupervised keyphrase extraction using self and cross attentions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* pp. 1919–1928

[9] Liu F, Liu F and Liu Y 2008 December Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion. In *2008 IEEE Spoken Language Technology Workshop*, pp. 181–184. IEEE

[10] Duari S and Bhatnagar V 2020 Complex network based supervised keyword extractor. *Expert Syst. Appl.* 140: 12876

[11] Santosh T Y S S, Sanyal D K, Bhowmick P K, Das P P 2020 Dake: Document-level attention for keyphrase extraction. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*. Springer International Publishing, pp. 392–401

[12] Akiko A 2003 An information-theoretic perspective of tf-idf measures. *Inf. Process. Manage.* 39(1): 45–65

[13] Umadevi M 2020 Document comparison based on TF-IDF metric. *Int. Res. J. Eng. Technol.* 7(02): 1546–1550

[14] Hashemzadeh B and Abdolrazzagh-Nezhad M 2020 Improving keyword extraction in multilingual texts. *Int. J. Electric. Comput. Eng.* 10(6): 2088–8708

[15] Boudin F 2013 A comparison of centrality measures for graph-based keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 834–838

[16] Florescu C and Caragea C 2017 February A position-biased pagerank algorithm for keyphrase extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 31, No. 1)

[17] Kamal S, Mita N and Suranjan G 2012 Machine learning based keyphrase extraction: comparing decision trees, naïve Bayes, and artificial neural networks. *J. Inf. Process. Syst.* 8(4): 693–712

[18] Zhang Q, Wang Y, Gong Y and Huang X-J 2016 November Keyphrase extraction using deep recurrent neural networks on twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 836–845

[19] Qinjun Q, Xie Z, Wu L and Li W 2019 Geoscience keyphrase extraction algorithm using enhanced word embedding. *Expert Syst. Appl.* 125: 157–169

[20] Gollam R, Saiful A, Mufti M, Zamli Kamal Z and Rahman Mohammed M 2020 Teket: a tree-based unsupervised keyphrase extraction technique. *Cogn. Comput.* 12: 811–833

[21] Litvak M and Last M 2008 August Graph-based keyword extraction for single-document summarization. In *Coling 2008: Proceedings of the Workshop Multi-source Multilingual Information Extraction and summarization*, pp. 17–24

[22] Dey P, Chaterjee A and Roy S 2019 Influence maximization in online social network using different centrality measures as seed node of information propagation. *Sādhanā* 44: 1–13

[23] Zhang J and Luo Y 2017 March Degree centrality, betweenness centrality, and closeness centrality in social network. In *2017 2nd International Conference on Modelling, Simulation and Applied Mathematics (MSAM2017)*. Atlantis Press, pp. 300–303

[24] Mark B 2004 Betweenness centrality in large complex networks. *Eur. Phys. J. B* 38(2): 163–168

[25] Kazuya O, Wei C and Li X-Y 2008 Ranking of closeness centrality for large-scale social networks. *Lect. Notes Comput. Sci.* 5059: 186–195

[26] Mihalcea R and Tarau P 2004 Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411

[27] Siddiqi S and Sharan A 2015 Keyword and keyphrase extraction techniques: a literature review. *Int. J. Comput. Appl.* 109(2)

[28] Wang L and Li S 2017 August PKU_ICL at SemEval-2017 task 10: Keyphrase extraction with model ensemble and external knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 934–937