Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 9, Issue. 9, September 2020, pg.83 – 93

Keyphrase Extraction from Document Using RAKE and TextRank Algorithms

J.S. Baruni[1]; Dr. J.G.R. Sathiaseelan[2]

[1] Department of Computer Science, Bishop Heber College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli, India

[2]Department of Computer Science, Bishop Heber College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli, India

[1] jsbaruni96@gmail.com

[2] jgrsathiaseelan@gmail.com

DOI: 10.47760/IJCSMC.2020.v09i09.009

Abstract— Traditional approaches to extract useful Keyphrase from a sentence rely heavily on human effort. In this paper, to overcome this challenge, Automatic Keyphrase Extraction algorithm has been used to extract a Keyphrase efficiently that reduces the scope for human errors and saves time. The Machine Learning algorithms detect the Keyphrase from a sentence that the user feeds as an input and sets a reminder using the Keyphrase. RAKE and Textrank algorithms help to extract Keyphrase or important terms of a given text document. RAKE and TextRank techniques applied to find and analyze the best possible way of extracting the Keyphrase efficiently. With slight modifications to the code, the algorithms can be implemented to serve different application domain such as message or threat decoding in military purposes and can be extended to use in speech-to-text translation and sentimental analysis of the data.

Keywords— Keyphrase Extraction, Approaches, Natural Language Processing, NLTK- POS Tagging, TF-IDF, RAKE, TextRank, Performance Analysis.

I. INTRODUCTION

Keyphrase extraction is a fundamental task in natural language processing that facilitates mapping of documents to a set of representative phrases[1].[2] The concise understanding of the text and grasping the central theme behind the given text can be achieved through Keyphrase extraction[3]. [4]Spending a huge amount of time in reading can be avoided. Information can be extracted efficiently comparing to the traditional extraction techniques.

At present times, where there exists a vast amount of information in the form of text on internet, the generation of Keyphrase has assumed much wider application and importance. [5]With the growing abundance of resource materials on the internet, the need of information retrieval calls for automatic tagging of a text or document to extract relevant information for a particular query of a user. Without any doubt, the task of manually tagging or summarizing such texts will be herculean and this calls for automation in this field to reduce the time and effort and of course to meet the unprecedented volume of information to be exchanged today. The rise of 'Big Data Analysis' will play a prominent role in phrase extraction.

Any key phrase model aims to generate words and phrases to summarize the given text. This paper contains various sections such as a section 1 is introduction, section 2 contains background work, section 3 discuss various approaches towards phrase Detection, section 4 divide into two subdivision, one explains Rapid automatic Keyphrase extraction and TextRank algorithm, section 5 shows performance analysis and finally section 6 provides conclusion.

II. BACKGROUND WORK

Keyphrase give a high-level description of a document's contents that is intended to make it easy for prospective readers to decide whether or not it is relevant for them [6]. Because Keyphrase summarize documents very concisely, they can be used as a low-cost measure of similarity between documents, making it possible to cluster documents into groups by measuring overlap between the Keyphrase they are assigned [7]. Automatic Keyphrase extraction is typically a two-step process: first, a set of words and phrases that could convey the topical content of a document are identified, then these candidates are scored or ranked and the "best" are selected as a document's Keyphrase. But they have other applications too. A related application is topic search: upon entering a Keyphrase into a search engine, all documents with this particular Keyphrase attached are returned to the user.

In summary, Keyphrase provide a powerful means for sifting through large numbers of documents by focusing on those that are likely to be relevant. In this light, we decided to construct an ensemble method for automatic keyword extraction. [8]Unsupervised Keyphrase extraction has a series of advantages over supervised methods. [9] Supervised Keyphrase extraction always requires the existence of a (large) annotated corpus of both documents and their manually selected Keyphrase to train on - a very strong requirement in most cases. [10] Supervised methods also perform poorly outside of the domain represented by the training corpus - a big issue, considering that the domain of new documents may not be known at all.

Unsupervised Keyphrase extraction [11] addresses such information constrained situations in one of two ways: (a) by relying on in-corpus statistical information (e.g., the inverse document frequency of the words), and the current document; (b) by only using information extracted from the current document. We employ the following unsupervised automatic Keyphrase extractors used for research documents such as TextRank and RAKE. In the following sections, we discuss how these automatic Keyphrase extractors work [12].

III. VARIOUS APPROACHES TOWARDS PHRASE DETECTION

- *Natural Language Processing-NLP*

NLP is the widely used technique to extract key phrases from large chunk of data. Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken [13]. NLP is a component of artificial intelligence (AI). Natural language refers to the way we humans communicate with each other namely, speech and text.

- *Term Frequency-inverse document frequency – TF-IDF*

The TF-IDF weight is a weight often used in information retrieval and text mining. Variations of the TF-IDF weighting scheme are often used by search engines in scoring and ranking a document's relevance given a query. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus (data-set)[14].

- *NLTK-POS Tagging*

NLTK- POS tagging is a supervised learning solution that uses features like the previous word, next word, is first letter capitalized etc. NLTK has a function to get POS tags and it works after tokenization process [15]. The dataset has to

be pre-processed before adding a tag. The following are the steps to implement POS tagging.

- Parsing of Text/ Sentence Segmentation:

  Text parsing is a common programming task that splits the given sequence of characters or values (text) into smaller parts based on some rules.

- Storing the segmented words/Sentence in List:

  The segmented word is then stored in a list. The sequence is further analyzed, tokenized and grammar is determined

- Tokenization:

  "Tokens" are usually individual words and "tokenization" is taking a text or set of text and breaking it up into its individual words. These tokens are then used as the input for other types of analysis or tasks, like parsing (automatically tagging the syntactic relationship between words).

- PART OF SPEECH(POS) Tagging:

  A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'

- Listing the Candidate Keyphrase**:**

  The candidate Keyphrase listed based on tags. The co-occurring Keyphrase are identified.

- Scoring the potential candidate Keyphrase:
  - The potential candidate Keyphrase are scored
  - The best Keyphrase are selected and scored.
  - From the given scores the models generate a Keyphrase.

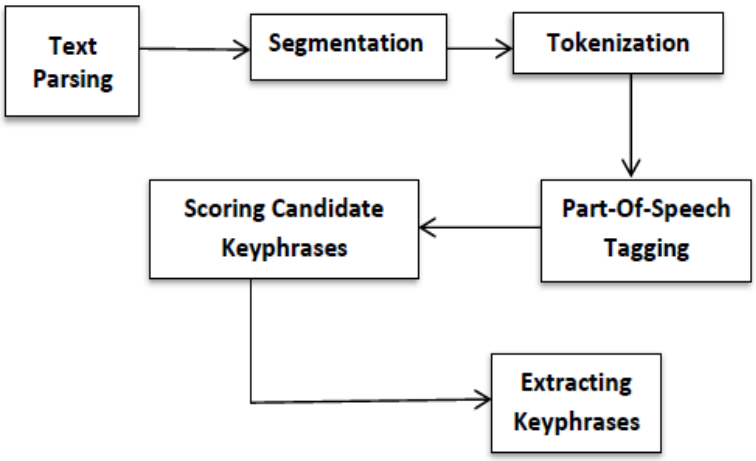


Fig 1. Phrase Detection pipeline

IV. KEYPHRASE EXTRACTION ALGORITHMS

In this paper, following sub sections contains workflow of Rapid Automatic Keyword Extraction (RAKE) which is an unsupervised, domain-independent, and language-independent method for extracting Keyphrase from individual documents[16][17] and workflow of TextRank algorithm is briefly discussed. The detail of the algorithm and its configuration parameters, and present results on a benchmark dataset of literature abstracts has been provided in the following sections.

A. Rapid Automatic Keyword Extraction-RAKE Algorithm

Rake refers to Rapid Automatic Keyphrase Extraction and it is efficient and fastest growing algorithm for keywords and Keyphrase extraction [18]. Candidates are extracted from the text by finding strings of words that do not include phrase delimiters or stop words (a, the, of, etc). This produces the list of candidate keywords/phrases.

A Co-occurrence graph is built to identify the frequency that words are associated together in those phrases. .A score is calculated for each phrase that is the sum of the individual word's scores from the co-occurrence graph. [19]An individual word score is calculated as the degree (number of times it appears + number of additional words it appears with) of a word divided by its frequency (number of times it appears), which weights towards longer phrases.

Adjoining keywords are included if they occur more than twice in the document and score high enough. An adjoining keyword is two keyword phrases with a stop word between them. [20][21]The top T keywords are then extracted from the content, where T is 1/3rd of the number of words in the graph. As below we visualize the text corpus that we created after pre-processing to get insights on the most frequently used words using RAKE algorithm.

B. TextRank Algorithm

In general, Text Rank creates a graph of the words and relationships between them from a document, then identifies the most important vertices of the graph (words) based on importance scores calculated recursively from the entire graph [22].

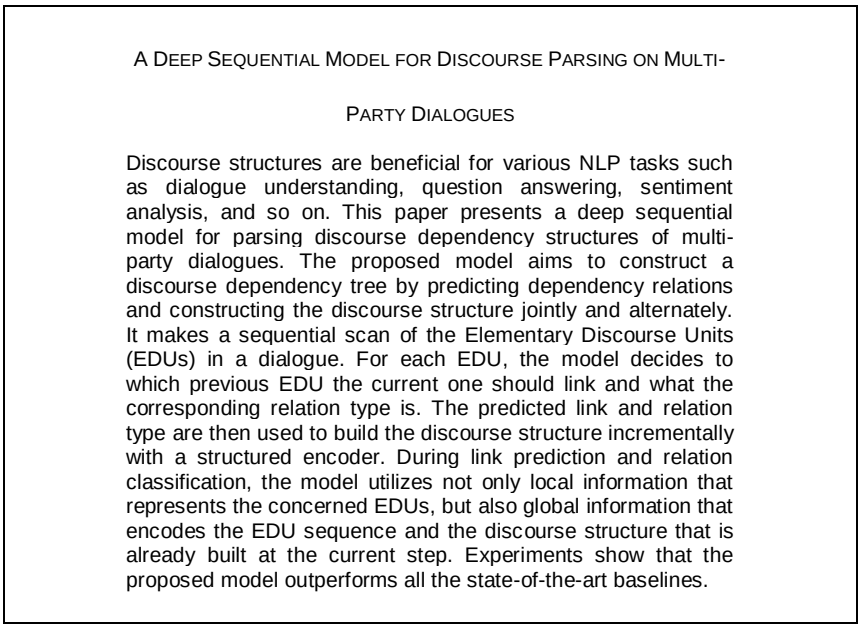
Candidates are extracted from the text via sentence and then word parsing to produce a list of words to be evaluated. The words are annotated with part of speech tags (noun, verb, etc) to better differentiate syntactic use. Each word is then added to the graph and relationships are added between the word and others in a sliding window around the word. [23]A ranking algorithm is run on each vertex for several iterations, updating all of the word scores based on the related word scores, until the scores stabilize – the research paper notes this is

typically 20-30 iterations. The words are sorted and the top N are kept (N is typically 1/3rd of the words). [24]

A post-processing step loops back through the initial candidate list and identifies words that appear next to one another and merges the two entries from the scored results into a single multi-word entry.[25] As below we visualize the text corpus that we created after pre-processing to get insights on the most frequently used words using TextRank algorithm.

V. PERFORMANCE ANALYSIS

The below shown fig 2 is one of the sample literature abstract extracted from Arxiv NLP papers with Github link. This abstract has been chosen randomly for Keyphrase evaluation using both RAKE and TextRank Keyphrase Extraction algorithm



A DEEP SEQUENTIAL MODEL FOR DISCOURSE PARSING ON MULTI-PARTY DIALOGUES

Discourse structures are beneficial for various NLP tasks such as dialogue understanding, question answering, sentiment analysis, and so on. This paper presents a deep sequential model for parsing discourse dependency structures of multi-party dialogues. The proposed model aims to construct a discourse dependency tree by predicting dependency relations and constructing the discourse structure jointly and alternately. It makes a sequential scan of the Elementary Discourse Units (EDUs) in a dialogue. For each EDU, the model decides to which previous EDU the current one should link and what the corresponding relation type is. The predicted link and relation type are then used to build the discourse structure incrementally with a structured encoder. During link prediction and relation classification, the model utilizes not only local information that represents the concerned EDUs, but also global information that encodes the EDU sequence and the discourse structure that is already built at the current step. Experiments show that the proposed model outperforms all the state-of-the-art baselines.

Fig 2. Sample Abstract to Extract Keyphrase

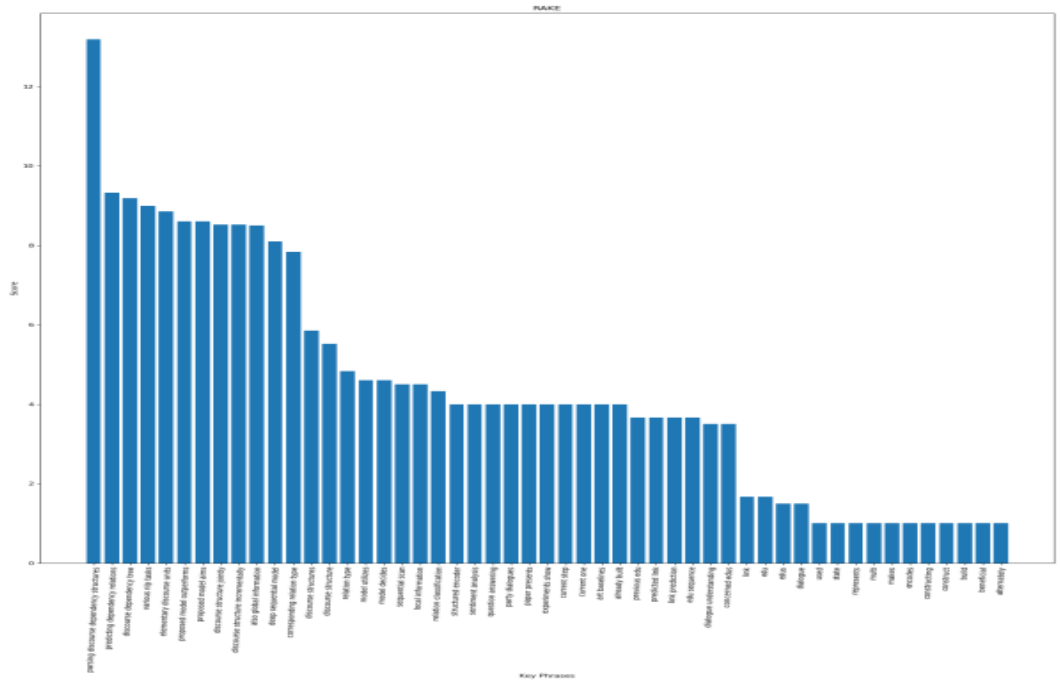
No's	Extracted Keyphrase using Rake Algorithm	Scores
1.	"parsing discourse dependency structures"	13
2.	"predicting dependency relations"	9.3
3.	"discourse dependency tree"	9.1
4.	"various nlp tasks"	9.0
5.	"elementary discourse units"	8.8
6.	"elementary discourse units"	8.6
7.	"proposed model outperforms"	8.5
8.	"discourse structure incrementally"	8.5
9.	"deep sequential model"	8.1
10.	" corresponding relation type"	7.8
11.	"discourse structures"	5.5
12.	"relation type"	4.8
13.	"model decides"	4.6
14.	"sequential scan"	4.5
15.	"sentiment analysis"	4.0
16.	"previous edu" "	3.6
17.	dialogue understanding"	3.5
18.	"link"	1.6
19.	"edu"	1.5
20.	"alternately"	1.0

Table 1. Extracted Keyphrase with scores using Rake algorithm.

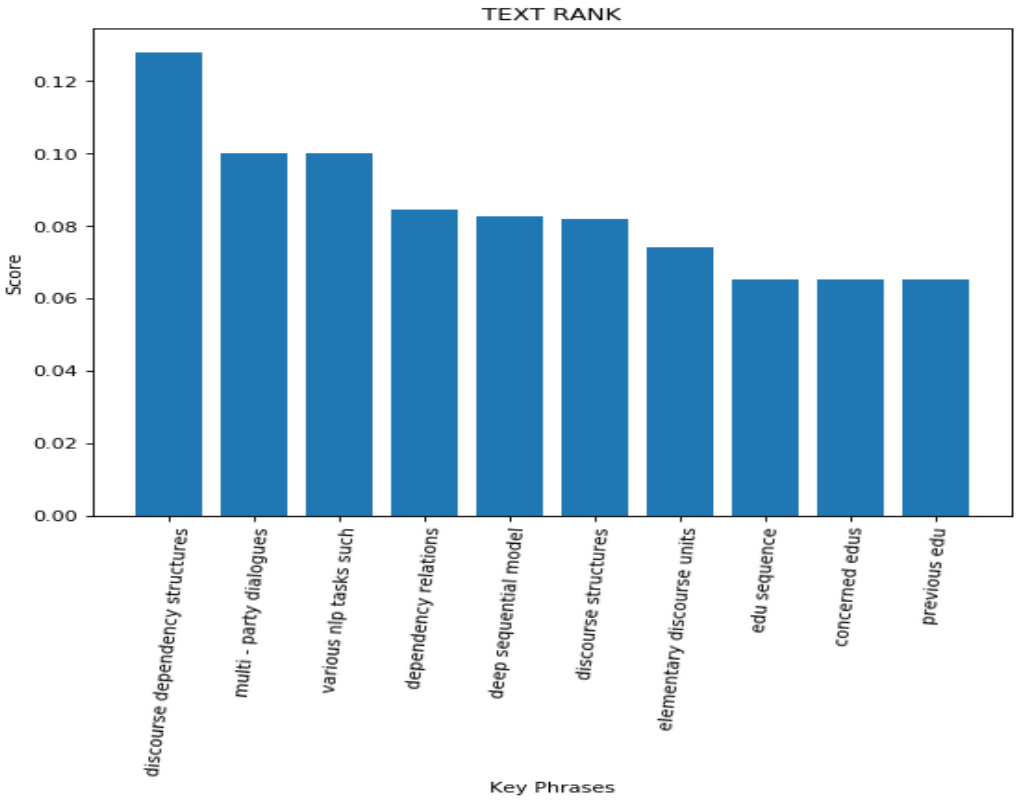
No's	Extracted Keyphrase using TextRank algorithm	Scores
1.	"Discourse Dependency Structures"	0.12
2.	"Multi-party dialogues"	0.10
3.	"Various nlp tasks such"	0.8
4.	"Dependency relations"	0.8
5.	"Deep sequential model"	0.8
6.	"Discourse structures"	0.8
7	"Elementary discourse units"	0.7
8.	"Edu sequence"	0.6
9.	"concerne edus"	0.6
10.	"Previous edu"	0.6

Table 2. Extracted Keyphrase with scores using TextRank algorithm.

Finally, we apply RAKE and TextRank algorithms to a corpus of research paper and define metrics for evaluating the exclusivity, essentiality, and generality of extracted Keyphrase, enabling a system to identify Keyphrase that are essential or general to document in the absence of manual annotations. From the above Table 1 showing that RAKE is more computationally efficient than TextRank shown in the Table 2 while achieving higher precision and comparable recall scores which we use to configure RAKE for specific domains and corpora. The most frequently Most frequently occurring N for RAKE and Textrank algorithms are shown below as grams unigrams, bi-grams and trigrams which clearly displays Keyphrase obtained with scores as shown below in graph 1 and graph 2.

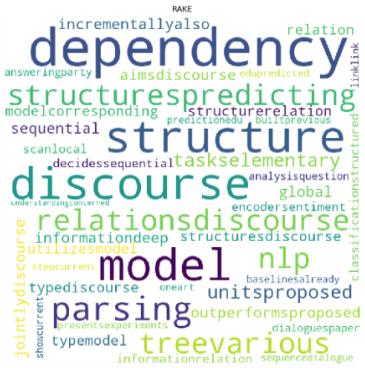


Graph 1. Most frequently occurring unigrams, bi-grams and trigrams using Rake algorithm



Graph 2: Most frequently occurring unigrams, bi- grams and trigrams using TextRank algorithm.

As below we visualize the text corpus that we created after pre-processing to get insights on the most frequently used words using RAKE algorithm and TextRank algorithm.



WordCloud of Rake Algorithm

The most important thing to notice here is that TextRank gives us Keyphrase only one entry has two words, the rest have only one word, while RAKE gives us phrases.

Wordcloud of TextRank algorithm

VI. CONCLUSION

The above proposed was implemented in Python=3.7 and used the NLTK toolkit to preprocess text. Keyphrase extraction techniques spare time and assets, by allows to consequently investigating huge arrangements of information in not more than seconds. Keyphrase extraction automatically extracting and classifying information from document which gives a keen and strong course of action, making it possible to separate text for a colossal degree and get speedy and exact results. In this paper we implemented Rapid Automatic Keyphrase Extraction and TextRank algorithms for data driven text and analyzed the predictions and accuracy which results as scores in the table 1 and 2. The top keywords from the contents are displayed to the user. We infer that RAKE algorithm gives the best results. RAKE tool is used to produce a list of candidate keywords or phrases and the score calculated for each phrase depending upon features

of the word and correlation among them. Adjoining keyword are included if they occur more than twice in the text and given a high score compare to TextRank algorithm.

REFERENCES

[1]. Lima Subramanian and R.S Karthik, "Keyword Extraction: A Comparative Study Using Graph Based Model And Rake" March 2017.

[2]. Ambar Dutta, Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Jharkhand, India, "A Novel Extension for Automatic Keyword Extraction", Volume 6, Issue 5, May 2016.

[3]. M. Uma Maheswari, Dr. J. G. R. Sathiaseelan. "Text Mining: Survey on Techniques and Applications", International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064, Volume 6 Issue 6, June 2017.

[4]. Said A. Salloum, Mostafa Al-Emran, Azza Abdel Monem, and KhaledShaalan, "Using Text Mining Techniques for Extracting Information from Research Articles", Chapter in Studies in Computational Intelligence, DOI: 10.1007/978-3-319-67056-0_18 January 2018.

[5]. Tayfun Pay, Stephen Lucci, James L. Cox, "An Ensemble of Automatic Keyword Extractors: TextRank, RAKE and TAKE", Computación y Sistemas, Vol. 23, No. 3, 2019.

[6]. Alzaidy. R., Caragea, C., Giles, C.L.: "Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents". In: Proceedings of The World Wide Web Conference, pp. 2551–2557. ACM, 2019.

[7]. DebanjanMahata, John Kuriakose, Rajiv Ratn Shah, and Roger Zimmermann "Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings".

[8]. Howard, Jeremy, & Ruder, Sebastian, Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146, 2018.

[9]. Sifatullah Siddiqi, AditiSharan, "Keyword and Keyphrase Extraction Techniques: A Literature Review", International Journal of Computer Applications (0975 – 8887) Volume 109 – No. 2, January 2015.

[10]. Meng, Rui, Yuan, Xingdi, Wang, Tong, Brusilovsky, Peter, Trischler, Adam, & He, Daqing. "Does Order Matter? An Empirical Study on Generating Multiple Keyphrases as a Sequence", arXiv preprint arXiv:1909.03590, 2019.

[11]. Isabella Gagliardi and Maria Teresa Artese, " Semantic Unsupervised Automatic Keyphrases Extraction by Integrating Word Embedding with Clustering Methods", June 2020.

[12]. Gollum Rabby, Saiful Azad1, Mufti Mahmud · Kamal Z. Zamli1, Mohammed MostafizurRahman "TeKET: a Tree-Based Unsupervised Keyphrase Extraction Technique, Cognitive Computational", Published online" March 2020.

[13]. Beltagy, I., Cohan, A., Lo, K.: "Scibert: pretrained contextualized embeddings for scientific text", 2019.

[14]. Sang-Woon Kim and Joon-Min Gil, "Research paper classifcation systems based on TF-IDF and LDA schemes", https://doi.org/10.1186/s13673-019-0192-7, August 2019.

[15]. Aparna Bulusu, Sucharita V, "Research on Machine Learning Techniques for POS Tagging in NLP", International Journal of Recent Technology and Engineering,(IJRTE), ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019.

[16]. Teng-Fei Li, Liang Hu, Jian-Feng Chu, Hong-Tu Li, and Chi, "An Unsupervised Approach for Keyphrase Extraction Using Within-Collection Resources" 2017.

[17]. Kamil Bennani-Smires, Claudiu Musat, Andreaa Hossmann, Michael Baeriswy, and Martin Jaggi, "Simple Unsupervised Keyphrase Extraction using Sentence Embeddings" October2018.

[18]. S. Anjali Nair, M. Meera, M.G. Thushara, "A Graph-Based Approach for keyword extraction from documents", Second International Conference on Advance Computational and Communication Paradigms", ICACCP 2019.

[19]. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. "Language models are unsupervised multitask learners. OpenAI Blog", 2019.

[20]. Yan Ying,Tan Qingping, Xie Qinzheng, Zeng Ping and Li Panpan, "A graph-based approach of automatic keyphrase extraction", Procedia Computer Science, vol. 107, pp. 248-255, 2017.

[21]. Gollapalli, S.D., & Caragea, C. "Extracting keyphrases from research papers using citation networks", 2014.

[22]. Rada Mihalcea and Paul Tarau Department of Computer Science University of North Texas "TextRank: Bringing Order into Texts".
[23]. Jinzhang Zhou, "Keyword extraction method based on word vector and TextRank", Application Research of Computers, 36, 5, 2019.
[24]. Suhan pan, Zhiqiang Li, Juan Dai, "An improved TextRank keywords extraction algorithm", ACM TURC '19: Proceedings of the ACM Turing Celebration Conference – China, May 2019.
[25]. Florescu, C., Caragea, and C.: Position Rank: an unsupervised approach to keyphrase extraction from scholarly documents. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 1105–1115, 2017.

