Computers, Materials & Continua
DOI:10.32604/cmc.2021.014330
Article

Tech Science Press

An Abstractive Summarization Technique with Variable Length Keywords as per Document Diversity

Muhammad Yahya Saeed[1], Muhammad Awais[1], Muhammad Younas[1], Muhammad Arif Shah[2,*], Atif Khan[3], M. Irfan Uddin[4] and Marwan Mahmoud[5]

[1]Department of Software Engineering, Government College University, Faisalabad, Faisalabad, 38000, Pakistan
[2]Department of IT and Computer Science, Pak-Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur, Pakistan
[3]Department of Computer Science, Islamia College Peshawar, Peshawar, Pakistan
[4]Institute of Computing, Kohat University of Science and Technology, Kohat, Pakistan
[5]Faculty of Applied Studies, King Abdulaziz University, Jeddah, Saudi Arabia
[*]Corresponding Author: Muhammad Arif Shah. Email: arif.websol@gmail.com

Received: 14 September 2020; Accepted: 10 October 2020

Abstract: Text Summarization is an essential area in text mining, which has procedures for text extraction. In natural language processing, text summarization maps the documents to a representative set of descriptive words. Therefore, the objective of text extraction is to attain reduced expressive contents from the text documents. Text summarization has two main areas such as abstractive, and extractive summarization. Extractive text summarization has further two approaches, in which the first approach applies the sentence score algorithm, and the second approach follows the word embedding principles. All such text extractions have limitations in providing the basic theme of the underlying documents. In this paper, we have employed text summarization by TF-IDF with PageRank keywords, sentence score algorithm, and Word2Vec word embedding. The study compared these forms of the text summarizations with the actual text, by calculating cosine similarities. Furthermore, TF-IDF based PageRank keywords are extracted from the other two extractive summarizations. An intersection over these three types of TD-IDF keywords to generate the more representative set of keywords for each text document is performed. This technique generates variable-length keywords as per document diversity instead of selecting fixed-length keywords for each document. This form of abstractive summarization improves metadata similarity to the original text compared to all other forms of summarized text. It also solves the issue of deciding the number of representative keywords for a specific text document. To evaluate the technique, the study used a sample of more than eighteen hundred text documents. The abstractive summarization follows the principles of deep learning to create uniform similarity of extracted words with actual text and all other forms of text summarization. The proposed technique provides a stable measure of similarity as compared to existing forms of text summarization.

Keywords: Metadata; page rank; sentence score; word2vec; cosine similarity

1 Introduction

It is challenging to process unstructured text documents without getting some prior idea about them, in the form of metadata. By applying text summarization techniques, the cost of text processing decreases, as the text mining algorithm utilizes only those documents which mainly relate to the text queries. Although these text queries rely solely on the metadata, the query results comprise over actual text documents [1,2]. Assessing multiple text documents is a time-consuming and challenging task. Therefore, metadata extraction techniques have a vital role in Text Mining [3]. The extractive summarization gives actual lines but does not contain the whole theme of the written text. Keywords extraction techniques extract crucial words but generally do not verify the significance of these words over actual or reduced extracted text [4,5]. Ultimately the efficiency of these information extraction techniques either decreases or increases as per content & context vulnerabilities of the underlying text [6]. Less efficient text assessment ultimately lacks to represent the real theme of long text, and it has chances to waste the applied effort during the query processing [7].

There are two broader ways of keywords-based metadata abstractive summarization. The first is single document keywords extraction, and the other is making a dictionary of keywords for multiple text documents [8]. Query processing utilizes the individual and collective metadata mainly in three different manners to assess a similar set of documents instead process the vast bulk of text documents. In the first method, the query matching process marks the related set of documents with single document metadata. In the second method, we match the individual document metadata with each other and group them as per metadata similarity criteria. This step facilitates to form a dictionary of unique keywords for each set of related documents. In this technique, a piece of text can fall into multiple subgroups. To improve the deficiencies of these two methods, we apply the clustering technique. In this method, the text clustering technique creates unique clusters from the keywords-based metadata. These clusters represent a specific group of documents [8,9]. The metadata processing fundamentally depends on the quality of the underlying abstractive text summarization technique.

In the second technique, the reduced text contains those potential lines of the text, which may represent the actual context of the long paragraph. There are two widely applied techniques for sentence extraction. In the first technique, we initially identify unique words and then calculate the words occurrence frequencies. We use these frequencies to assign sentence scores to the various lines of the text document. Ultimately, high score lines become visible and extracted to represent the actual text. The process applied in this sort of text extraction mainly follows the Sentence Score Algorithm (SSA) [10,11]. In the second method, we follow the word embedding principles referred to as the Word2Vec model. In this model, instead of calculating the terms occurrence frequencies, the frequencies of their mutual occurrences play the leading role, e.g., bread & butter, tea & sugar. In the current paper, we have applied Python language package Gensim for word embedding [11,12]. By using this package, we have applied the Gensim Based Word2Vec Algorithm (GW2VA) to create text summaries.

Natural Language Processing (NLP) has numerous real-life applications that require text mining in identifying the text necessary [3,7]. In these applications, some issues regarding metadata processing exist like length, type, diversity [9,13,14,15]. Our study jointly applies multiple summarizations to address the following problems.

Improved Parallel *Corpus* Analysis: We have created an enriched *corpus* besides just creating the keyword-based assessment of the text. This technique has a single time *corpus* creation mechanism, with multiple benefits. The first benefit is the option of cross similarity verification and ease of switching between numerous forms of metadata. This *corpus* creation technique involves a one-time effort and reduces the query processing cost with better accuracy. Secondly, it relies on four main *corpus* processing areas, combined under one method and these are single Document Summarization (SDS), Multiple Document Summarization (MDS), abstractive summarization, and extractive summarization.

How Many Keywords Are Sufficient to Extract: The previous approaches of text extraction mostly extract the fixed-length TF-IDF based PRK from a paragraph. Usually, there exist no specific rules to fix the number of keywords from a long article [7,11,16]. We have designed an approach which gives the variable-length keywords for each paragraph.

Applicable to Previous Studies: This study applies to all those studies which have used a keywords-based metadata approach in text mining. We have presented the comparative results in the Section 5 of this paper.

Unique Features: The presented study is different from other keywords extraction studies. The previous studies do not target the information diversity in the multiple documents of a given *corpus* [5,14,17]. The current technique follows the principles of deep learning. It interrelates the number of keywords of a paragraph with the diversity of its information, i.e., For more diverse text, our technique generates more keyword, and for the less unalike document, the keywords are also fewer. This keyword extracting technique attempts to assure that every paragraph has an equally capable set of keywords to present the actual theme of the article. We have discussed the multiple applied techniques in Section 3 of this paper. We have also explained every step of our technique in Section 4.

This section gives an overview of the presented work, scope and need of the paper. Rest of the paper is organized as follows: Section 2 presents the related work. Section 4 detailed description about implemented steps, with the summary. Section 5 step by step briefing of the proposed technique with tables and graphs, with the summary. Section 6 conclusion of the entire work.

2 Related Work

Qaiser and Ali explored the use of TF-IDF and examined the relevance of keywords in the text documents. Their research focused on how this algorithm process several unstructured documents. First, the algorithm designed to implement TF-IDF then to test the results of the algorithm with the strengths and weaknesses of the TD-IDF algorithm. They discussed the way to fix the flaws of text relevance and their effects with future directions of research. They suggested that the TF-IDF algorithm is simple to implement, but it has limitations. In today's big data world, text mining requires new data processing techniques. e.g., they discussed a variant of TF-IDF applied at the inter-language level using statistical translation. They proved that genetic algorithms improve TF-IDF. They stated about the search engine giants Google had adapted these algorithms of PageRank to display the relevant results when a user places a query. TF-IDF can work with other methods such as Naive Bayes to get better results.

Roul et al. performed a classification by using the document contents and its reference structure. Their proposed model combined the PageRank with TF-IDF. Their results improved the document relevance to its PageRank keywords. They showed the idea to fix the reference structure based on the similarity of the document and proved that the proposed classification method is promising and guide to auto-classification. For this, they developed a classification model and a series of page types that recreate the link structure. By doing this, they combined the advantages of TF-IDF with the PageRank algorithm. This work has three goals. First, it gives the direction of the proposed classification model to allows us to understand the effectiveness of this approach. Secondly, it makes a comparison with other known standard classification models. Third, query processed by a combination of different parameters.

Ganiger and Rajashekharaiah focused the automatic text reduction area of text mining. They divided summaries into short-text passages and phrases. The primary purpose of their work was to obtain a concise and informative text from the source document. They used standard algorithms for extracting keywords and applied TF-IDF as a baseline algorithm. In their work, they performed the training for keyword extraction algorithms. Their keyword extraction algorithm consisted of multiple parts to compare and evaluate the accuracy and completeness of the applied algorithm. They showed that TF-IDF is the main algorithm for generating suitable keywords.

The work of Yi-ran and Meng-Xin based on a words network extraction method that ignores useless characters. They created a network that does not include all existing keywords. They also proved that the classical algorithms belonging to this area contribute more complexity. In their work, they proposed a keywords extraction method based on the PageRank algorithm. Their proposed algorithm created weights for the common word and divided these words by the corresponding value of the word frequency. By determining the position of the weighting factor, the importance of each word shown.

Pan et al. worked over keyword extraction and used it in the keyword's assessment methodology. They focused on the issue of how to use keywords quickly and accurately in text processing. They showed that there exist many options to use keywords in many ways to improve the accuracy and flexibility of the text extraction process. In their work, they extracted the words through the improved algorithm for the TextRank keywords based on the TF-IDF algorithm. They applied estimation algorithms to calculate the importance of words in the text-based results. They further based the execution of the TextRank algorithm on these results. Finally, they used these keywords to perform the deletion of unnecessary keywords. Their findings have shown that their method of extracting the keywords is more accurate than traditional TF-IDF and text classification methods.

Li and Zhao proved that sentence score summaries are often less useful in email text filtration due to sending and receiving short character messages, as these become common compound keywords. They stated that besides using sentence scores, the classification algorithm based on graphs give better keywords assessment. They implemented graphs by building a vector of concepts and by measuring the similarity of the words. Finally, they constructed an array of keywords and extracted keywords by using the TextRank keyword. As compared to the traditional TextRank algorithm, their algorithm worked more effectively and provided the text extraction by the conventional TextRank and TF-IDF algorithm.

Mahata et al. used an unsupervised approach that combines PageRank and neural network algorithm. In their method, they worked with text documents using embedded keywords. The main application of their model was to select a set of keywords relating to the defined similarity estimates. They used two related datasets to apply keyword ranking and to consider the text summary suitability. In their work, they tried to find the concepts in the content of the text document and assigned weights to the candidate words. Their proposed system based on a set of experimental data to evaluate all candidate PageRank keywords. They suggested their work for multimodal dataset to extract keywords relating to images and tagging them by automatic indexing.

3 Description of the Applied Techniques

The text assessment process applies various text mining techniques simultaneously. Some of these techniques have their role in almost every type of *corpus* processing, e.g., the text pre-processing is the first step [11,13]. After this, the next text mining processes are word identifications, word types assessments, stemming, lemmatization, etc., [12,15]. We have discussed these techniques in this section, and all of these relate to the current study.

3.1 Text Pre-Processing

Text Pre-Processing techniques cleanse the dataset by eliminating useless tokens. It is a widely accepted practice in all sorts of Natural Language Processing. (NLP) projects. This phase removes special symbols and all those elements which add noise to the text [7,13,15]. There exists no specific definition of text-noise, but generally, anything which causes unnecessary text processing is noise, e.g., inconsistent data, duplication, wrong writings, etc., Besides removing the text elements, there are various other forms of the text processing like case folding. Case folding represents multiple text elements in a specific kind, e.g., it may represent proper nouns as capital words and remaining text as small letters, etc., This sort of text processing makes the terms recognition easy for the text engineer and search engine [14,16,17].

3.2 Tokenization

The critical purpose of tokenization is dividing the sentence into parts, which termed as tokens, and chucking away many words like punctuation, etc., Tokenization is the technique of distributing the data into the form of words, symbols, or phrases [4,17,18]. The straightforward drive of tokenization is to classify strong word arguments. Some tasks necessity relates to tokenization like the nature of language, like, English and Chinese languages are different regarding the use of the white spaces. Similarly, there are certain words like compound words, slangs, words joined with symbols, etc., All these types of terms need unique tokenization routines in the process of text processing [13,17,19].

3.3 Stop Words Removal

NLP has many common words that have little significance in the semantic context, and we name them as stop words. There exist various lists of stop words in every text processing system [5,17,20]. These words are essential for the reader as these words give the sense to the sentences. But these words are not always useful for text processing, and search engines do not use these words for the findings of the relevant results. These words are like 'and,' 'or,' this,' is,' etc., These words frequently occur in all text documents and create hurdles in text processing. Removal of these words exists as a typical text processing routine in the cases, where these words have no role in the underlying text mining [17,21,22]. After removing these words from the text, the overhead of text processing also becomes reduced [21].

3.4 Parts of Speech Tagging

In Part of Speech (POS), we perform recognition of nouns, verbs, proverbs, objectives, etc., we can also represent POS as POS-tagging. In this process, the NLP process creates the annotations of different types with their type's identification [13,16,19]. This process performs the grammatical tagging and helps to understand the connections of the sentence and their relations. This form of token labelling provides the enriched metadata for the text query processing [21,23,24].

3.5 Stemming and Lemmatization

During the Stemming process, we try to identify the word's base or stem, and we remove the affixes. We use this step to replace the word, and by doing this, the specific term comes to its original root. There are various examples like 'eat,' 'eating.' By stemming, we reduce 'eating' to 'eat' as 'eat' is the stem. Any line referring to these words can be related to the context where the food items discussed [23,25]. Stemming has a related process, which we call as Lemmatization or Lemming. This process assures that we reduce the words to their exact stems. If we have two words 'care,' 'car' then we cannot convert 'care' to 'car,' as 'car' is not stem to the word 'care' and so on [15,22].

3.6 Term Frequency-Inverse Document Frequency

Cleansed text after the preprocessing serves as an input to perform the text summarization process. We calculate TF-IDF for a text *corpus*, and it provides numerical value, which shows the importance of the word to a document. TF-IDF values increase as per the word's repetition proportionally for a given document. The frequency of the words in the text *corpus* balance this value [14,27]. This technique has wide existence of its use over the internet as it differentiates the common and uncommon words in the text *corpus*. Term Frequency (TF) provides the raw frequency of the word in the document. Inverse Document Frequency (IDF) helps to assess the word, whether the word is common in the document or uncommon, etc., First, we calculate the total documents. Then we calculate the documents containing the word concerned. We divide these numbers to assess documents ratio containing the term [15,28]. Following is the formula of TF-IDF.

$$TF(tr, d) = \frac{tr_i}{\sum_k tr_k}. \tag{1}$$

$$IDF(tr, d) = \log \frac{N}{|\{d \in D \ : t \in \mathbf{d}\}|} \tag{2}$$

where tr_i = Term tr in Document d and

$\sum_k tr_k$ = Total Words of Document d

N = Total number of *Corpus* documents: N = |D|

3.7 Page Rank Keywords

We use the Page Rank (PR) algorithm for processing the text documents, which we link or hyperlink like the web pages. This algorithm works on the principle of assigning each paragraph a numerical value in such a way that this value implies the importance of a document as compared to the other materials [11,17,29]. This algorithm works satisfactorily for all those documents which have reciprocal links. We use the PR value for the given page's significance indication. The PR value bis on two parameters, i.e., the total number of PR pages and the calculated values of the PR for these pages [7,13,30]. Consider the pages W, X, Y have a link to the page Z. If we want to calculate the PR value for Z, then it will take the sum of all PR values of W, X, and Y.

3.8 Sentence Scoring

Sentence scoring is the process that assigns the score to the paragraph lines. This score basis on the frequency of the words in the given paragraph. This process has multiple variants over the Internet. Sentence scoring relies either on predefined keywords or post analyzed keywords. In the case of predefined words, lines containing these specified words have a higher score than other lines [22,31]. There exists customized search engines or software applications which highlight the query-specific words in these query-specific lines. We may extract these lines, as per the requirement. In a second way, after performing the document analysis, the high-frequency words facilitate selecting the crucial lines of the text. This latter method of sentence scoring has a common application as the extractive summarization method [28,32].

3.9 Word Embedding

We apply the Word2Vec model for the context assessment, and we train this model over the textual data. The fundamental objective of this model is mapping the sentence words as hooked on a small dimensional vector space. This model identifies the terms of a close relation. It maintains the distance of these words closer and smaller by keeping view of the context or meanings of these words [15,28]. We train this model by artificial intelligence, and we use the neural networks over the vector words, in predicting the document context. The outcome of this processing depends on generating similar context words in close vectors [14,27]. We obtain word clusters, which are more probable to occur together simultaneously. This model keeps improving over time as it is time-variant, i.e., the words which were occurring together in the last decade are not necessarily closer to each other in the current decade [17,29].

3.10 Cosine Similarity

We use the Cosine Similarity (CS) to measure the resemblance among the various documents. It endorses maximum probable articles of interest for the user. Item resemblance approbation is subject to the CS value [28,30]. CS serves as an optimal choice when the documents have high-dimensional attributes, particularly in retrieving text for the information analysis. We use CS for similarity calculation among both the items & users, in the form of item & user vector [7,18,22]. CS formula is presented in Eq. (3).

$$CS(a,\ b) = \frac{\sum_{j=1}^{k} a_j b_j}{\sqrt{\sum_{j=1}^{k} a_j^2}\sqrt{\sum_{j=1}^{k} b_j^2}} \tag{3}$$

In this section, we have briefly described our utilized techniques. These techniques perform the text analysis and provide the base for the metadata generation. The text analysis uses the study of the keyword in various forms. We have described the text preprocessing as the primary step of NLP. To obtain the main keywords of the paragraph, we apply 'stop words' removal. Then we apply the POS-tagging to differentiate the word types. We group the words to their stems and analyze them regarding their context. We have discussed the tokenization process used in abstractive summarization by TF-IDF with the PRK. We have discussed sentence scoring and Word embedding, used in the extractive summary. We have precisely mentioned CS, as we applied it in the current study to assess the similarity between the various forms of summarization.

4 *Corpus* Creations, Processing, and Extraction Metadata

In this paper, we have presented the way to select improved keywords from the actual text and its multiple reduced forms. This technique gives enhanced *corpus* view and selects the different number of keywords as per the diversity of numerous types of *corpus*. We have explained all the steps of our technique form the Sections 4.1–4.6 and the block diagram of these steps depicted in Fig. 1.

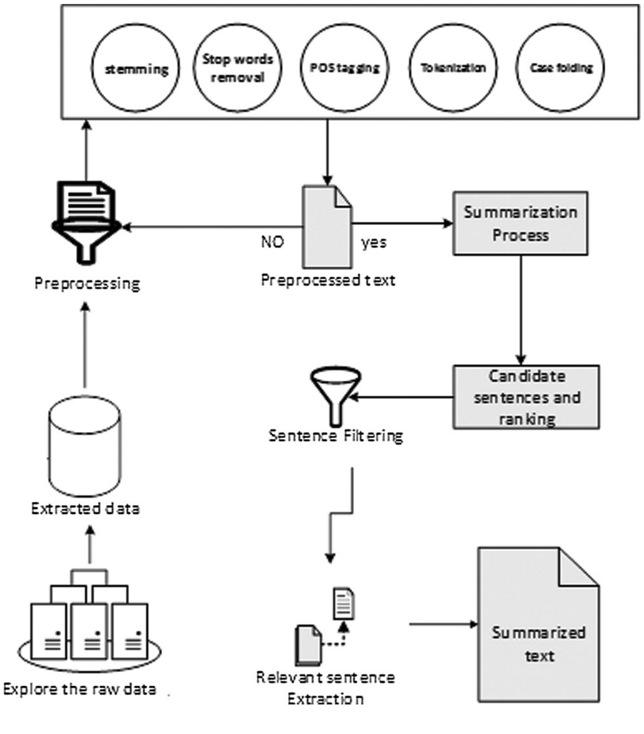


Figure 1: Text mining process from tokenization to summarized text extraction

4.1 *Preprocessing of Text*

We have preprocessed the text to remove its abnormalities. We have utilized this text in multiple summarizations of our proposed technique. We have performed it in the following steps.

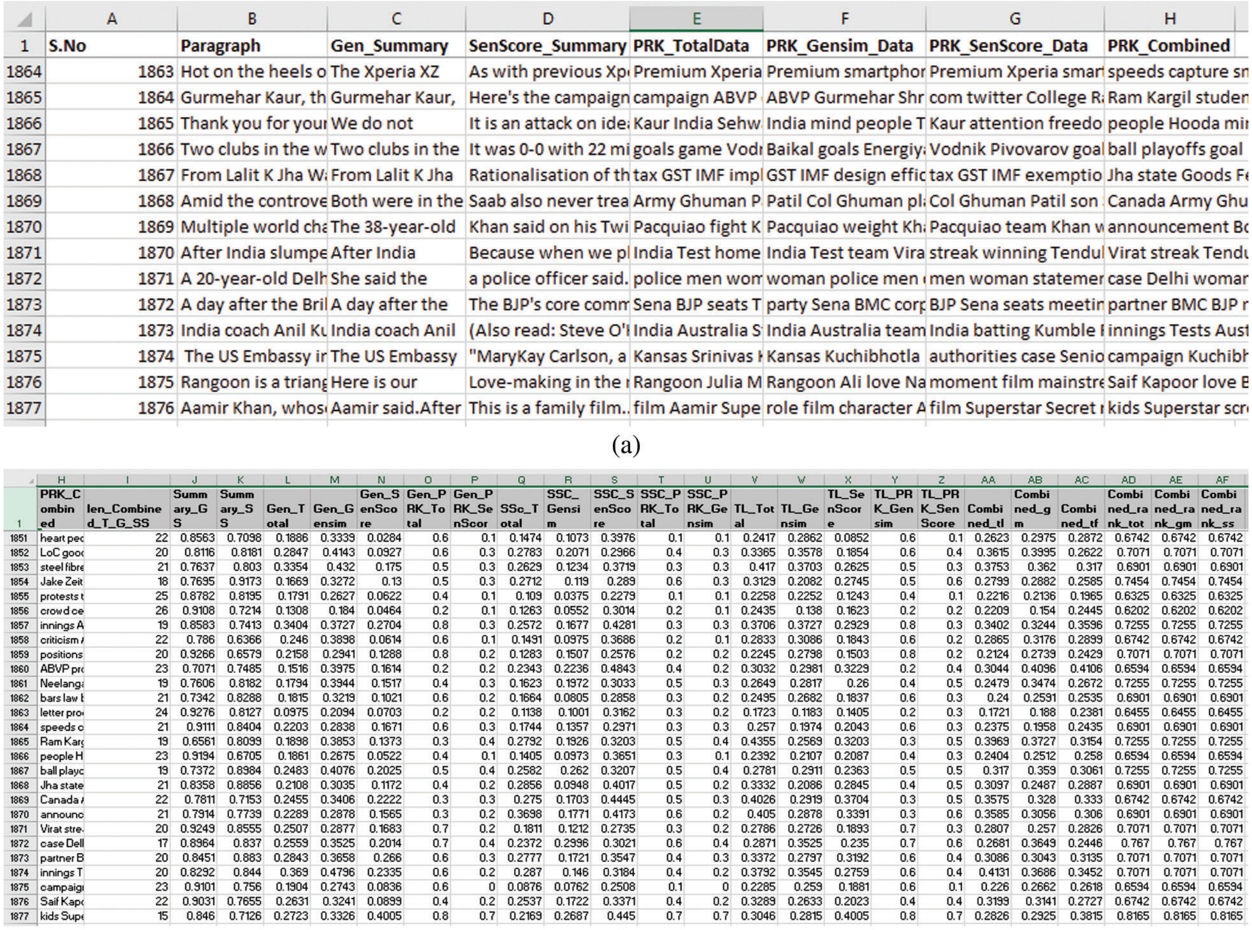
- We have removed the non-ASCII character and unnecessary numerical values from the text.
- We have identified the named entities and abbreviations by using the dictionary approach.
- These steps provided the cleansed text, and we have applied two extractive summarizations over this sentence-based text.

- Tokenization has critical importance in morphological analysis of the text. We have applied tokenization and further steps by using Python language package NLTK.
- The stop word removal performed over the cleansed text.
- Next, we have performed the POS tagging over the cleansed text.
- Then the stemming and lemmatization process applied to reduce the text further.
- Our abstractive summarization technique basis on this token-based cleansed text.

4.2 *Word Embedding Based Extractive Text Summarization*

Word embedding refers to a set of techniques used for modelling the language, and these techniques have NLP learning features. We apply vocabulary to map the tokens of words and phrases to the real number vectors. We have used Python language package Gensim to obtain Word2Vec summaries of the dataset [6,19]. We have performed it as below.

- This summarization performed over individual paragraphs of the dataset.
- The obtained summaries placed in an adjacent column with the actual text paragraph, see Fig. 2(a).
- The text on average is one third reduced in each paragraph by this summarization technique.
- These summaries rely on those words which have high frequency to occur together in various documents.



	A	B	C	D	E	F	G	H
1	S.No	Paragraph	Gen_Summary	SenScore_Summary	PRK_TotalData	PRK_Gensim_Data	PRK_SenScore_Data	PRK_Combined
1864	1863	Hot on the heels o	The Xperia XZ	As with previous Xp	Premium Xperia	Premium smartphor	Premium Xperia smar	speeds capture sn
1865	1864	Gurmehar Kaur, th	Gurmehar Kaur,	Here's the campaign	campaign ABVP	ABVP Gurmehar Shr	com twitter College R	Ram Kargil studen
1866	1865	Thank you for your	We do not	It is an attack on ide	Kaur India Sehw	India mind people T	Kaur attention freedo	people Hooda mi
1867	1866	Two clubs in the w	Two clubs in the	It was 0-0 with 22 mi	goals game Vodr	Baikal goals Energiy	Vodnik Pivovarov goal	ball playoffs goal
1868	1867	From Lalit K Jha W	From Lalit K Jha	Rationalisation of th	tax GST IMF impl	GST IMF design effic	tax GST IMF exemptio	Jha state Goods F
1869	1868	Amid the controve	Both were in the	Saab also never trea	Army Ghuman P	Patil Col Ghuman pl	Col Ghuman Patil son	Canada Army Ghu
1870	1869	Multiple world cha	The 38-year-old	Khan said on his Twi	Pacquiao fight K	Pacquiao weight Kh	Pacquiao team Khan w	announcement Bo
1871	1870	After India slumpe	After India	Because when we pl	India Test home	India Test team Vira	streak winning Tendul	Virat streak Tendu
1872	1871	A 20-year-old Delh	She said the	a police officer said.	police men wor	woman police men	men woman statemer	case Delhi womar
1873	1872	A day after the Bri	A day after the	The BJP's core comr	Sena BJP seats T	party Sena BMC corp	BJP Sena seats meetir	partner BMC BJP r
1874	1873	India coach Anil Ku	India coach Anil	(Also read: Steve O'	India Australia S	India Australia team	India batting Kumble	innings Tests Aust
1875	1874	The US Embassy ir	The US Embassy	"MaryKay Carlson, a	Kansas Srinivas	Kansas Kuchibhotla	authorities case Senio	campaign Kuchibh
1876	1875	Rangoon is a triang	Here is our	Love-making in the	Rangoon Julia M	Rangoon Ali love Na	moment film mainstre	Saif Kapoor love B
1877	1876	Aamir Khan, whos	Aamir said.After	This is a family film..	film Aamir Supe	role film character A	film Superstar Secret	kids Superstar scr

(a)

	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1	PRK_Combined	len_Combined_T_G_SS	Summary_GS	Summary_SS	Gen_Total	Gen_Gensim	Gen_SenScore	Gen_PRK_Total	Gen_PRK_SenScor	SSc_Total	SSC_Gensim	SSC_SenScore	SSC_PRK_Total	SSC_PRK_Gensim	TL_Total	TL_Gensim	TL_SenScore	TL_PRK_Gensim	TL_PRK_SenScore	Combined_tl	Combined_gm	Combined_tf	Combined_rank_tot	Combined_rank_gm	Combined_rank_ss
1851	heart pec	22	0.8563	0.7098	0.1886	0.3339	0.0284	0.6	0.1	0.1474	0.1073	0.3976	0.1	0.1	0.2417	0.2862	0.0852	0.6	0.1	0.2623	0.2975	0.2872	0.6742	0.6742	0.6742
1852	LoC gooc	20	0.8116	0.8181	0.2847	0.4143	0.0927	0.6	0.3	0.2783	0.2071	0.2966	0.4	0.3	0.3365	0.3578	0.1854	0.6	0.4	0.3615	0.3995	0.2622	0.7071	0.7071	0.7071
1853	steel fibre	21	0.7637	0.803	0.3354	0.432	0.175	0.5	0.3	0.2629	0.1234	0.3719	0.3	0.3	0.417	0.3703	0.2625	0.5	0.3	0.3753	0.362	0.317	0.6901	0.6901	0.6901
1854	Jake Zeit	18	0.7695	0.9173	0.1669	0.3272	0.13	0.5	0.3	0.2712	0.119	0.289	0.6	0.3	0.3129	0.2082	0.2745	0.5	0.6	0.2799	0.2882	0.2585	0.7454	0.7454	0.7454
1855	protests t	25	0.8782	0.8195	0.1791	0.2627	0.0622	0.4	0.1	0.109	0.0375	0.2279	0.1	0.1	0.2258	0.2252	0.1243	0.4	0.1	0.2216	0.2136	0.1965	0.6325	0.6325	0.6325
1856	crowd ce	26	0.9108	0.7214	0.1308	0.184	0.0464	0.2	0.1	0.1263	0.0552	0.3014	0.2	0.1	0.2435	0.138	0.1623	0.2	0.2	0.2209	0.154	0.2445	0.6202	0.6202	0.6202
1857	innings A	19	0.8583	0.7413	0.3404	0.3727	0.2704	0.8	0.3	0.2572	0.1677	0.4281	0.3	0.3	0.3706	0.3727	0.2929	0.8	0.3	0.3402	0.3244	0.3596	0.7255	0.7255	0.7255
1858	criticism /	22	0.786	0.6366	0.246	0.3898	0.0614	0.6	0.1	0.1491	0.0975	0.3686	0.2	0.1	0.2833	0.3086	0.1843	0.6	0.2	0.2865	0.3176	0.2899	0.6742	0.6742	0.6742
1859	positions	20	0.9266	0.6579	0.2158	0.2941	0.1288	0.8	0.2	0.1283	0.1507	0.2576	0.2	0.2	0.2245	0.2798	0.1503	0.8	0.2	0.2124	0.2739	0.2429	0.7071	0.7071	0.7071
1860	ABVP prc	23	0.7071	0.7485	0.1516	0.3975	0.1614	0.2	0.2	0.2343	0.2236	0.4843	0.4	0.2	0.3032	0.2981	0.3229	0.2	0.4	0.3044	0.4096	0.4106	0.6594	0.6594	0.6594
1861	Neelanga	19	0.7606	0.8182	0.1794	0.3944	0.1517	0.4	0.3	0.1623	0.1972	0.3033	0.5	0.3	0.2649	0.2817	0.26	0.4	0.5	0.2479	0.3474	0.2672	0.7255	0.7255	0.7255
1862	bars law b	21	0.7342	0.8288	0.1815	0.3219	0.1021	0.6	0.2	0.1664	0.0805	0.2858	0.3	0.2	0.2495	0.2682	0.1837	0.6	0.3	0.24	0.2591	0.2535	0.6901	0.6901	0.6901
1863	letter pro	24	0.9276	0.8127	0.0975	0.2094	0.0703	0.2	0.2	0.1138	0.1001	0.3162	0.3	0.2	0.1723	0.1183	0.1405	0.2	0.3	0.1721	0.188	0.2381	0.6455	0.6455	0.6455
1864	speeds c	21	0.9111	0.8404	0.2203	0.2838	0.1671	0.6	0.3	0.1744	0.1357	0.2971	0.3	0.3	0.257	0.1974	0.2043	0.6	0.3	0.2375	0.1958	0.2435	0.6901	0.6901	0.6901
1865	Ram Karg	19	0.6561	0.8099	0.1898	0.3853	0.1373	0.3	0.4	0.2792	0.1926	0.3203	0.5	0.4	0.4355	0.2569	0.3203	0.3	0.5	0.3969	0.3727	0.3154	0.7255	0.7255	0.7255
1866	people H	23	0.9194	0.6705	0.1861	0.2675	0.0522	0.4	0.1	0.1405	0.0973	0.3651	0.3	0.1	0.2392	0.2107	0.2087	0.4	0.3	0.2404	0.2512	0.258	0.6594	0.6594	0.6594
1867	ball playc	19	0.7372	0.8984	0.2483	0.4076	0.2025	0.5	0.4	0.2582	0.262	0.3207	0.5	0.4	0.2781	0.2911	0.2363	0.5	0.5	0.317	0.359	0.3061	0.7255	0.7255	0.7255
1868	Jha state	21	0.8358	0.8856	0.2108	0.3035	0.1172	0.4	0.2	0.2856	0.0948	0.4017	0.5	0.2	0.3332	0.2086	0.2845	0.4	0.5	0.3097	0.2487	0.2887	0.6901	0.6901	0.6901
1869	Canada /	22	0.7811	0.7153	0.2455	0.3406	0.2222	0.3	0.3	0.275	0.1703	0.4445	0.5	0.3	0.4026	0.2919	0.3704	0.3	0.5	0.3575	0.328	0.333	0.6742	0.6742	0.6742
1870	announc	21	0.7914	0.7739	0.2289	0.2878	0.1565	0.3	0.2	0.3698	0.1771	0.4173	0.6	0.2	0.405	0.2878	0.3391	0.3	0.6	0.3585	0.3056	0.306	0.6901	0.6901	0.6901
1871	Virat stre.	20	0.9249	0.8555	0.2507	0.2877	0.1683	0.7	0.2	0.1811	0.1212	0.2735	0.3	0.2	0.2786	0.2726	0.1893	0.7	0.3	0.2807	0.257	0.2826	0.7071	0.7071	0.7071
1872	case Dell	17	0.8964	0.837	0.2559	0.3525	0.2014	0.7	0.4	0.2372	0.2996	0.3021	0.6	0.4	0.2871	0.3525	0.235	0.7	0.6	0.2681	0.3649	0.2446	0.767	0.767	0.767
1873	partner B	20	0.8451	0.883	0.2843	0.3658	0.266	0.6	0.3	0.2777	0.1721	0.3547	0.4	0.3	0.3372	0.2797	0.3192	0.6	0.4	0.3086	0.3043	0.3135	0.7071	0.7071	0.7071
1874	innings T	20	0.8292	0.844	0.369	0.4796	0.2335	0.6	0.2	0.287	0.146	0.3184	0.4	0.2	0.3792	0.3545	0.2759	0.6	0.4	0.4131	0.3686	0.3452	0.7071	0.7071	0.7071
1875	campaigi	23	0.9101	0.756	0.1904	0.2743	0.0836	0.6	0	0.0876	0.0762	0.2508	0.1	0	0.2285	0.259	0.1881	0.6	0.1	0.226	0.2662	0.2618	0.6594	0.6594	0.6594
1876	Saif Kapc	22	0.9031	0.7655	0.2631	0.3241	0.0899	0.4	0.2	0.2537	0.1722	0.3371	0.4	0.2	0.3289	0.2633	0.2023	0.4	0.4	0.3199	0.3141	0.2727	0.6742	0.6742	0.6742
1877	kids Supe	15	0.846	0.7126	0.2723	0.3326	0.4005	0.8	0.7	0.2169	0.2687	0.445	0.7	0.7	0.3046	0.2815	0.4005	0.8	0.7	0.2826	0.2925	0.3815	0.8165	0.8165	0.8165

(b)

Figure 2: Preview of the actual and summarized corpuses

4.3 Sentence Score Based Extractive Text Summarization

Text summarization through the sentence score includes four things, i.e., text pre-processing which we have discussed previously involve, identifying high-frequency words, ranking the sentences as per the score/frequency of identified words, ranked lines extraction in the form of an extracted summary [21,26]. This process has the following steps.

- This summarization performed over individual actual paragraphs of the dataset.
- The generated summaries placed in the adjacent column with Word2Vec reduced paragraph, see Fig. 2(a).
- The text of each paragraph reduced one third on average.
- These summaries utilize those words, which have a high frequency in the given document.

4.4 PRK Based Abstractive Text Summarization

Word We have applied PRK over the text documents. Generally, eight to ten keywords satisfactorily define the theme of the document. In search engine optimization techniques, a thousand words web page or blog have five to eight keywords, depending on cost per click criteria [31,33]. There exist no specific rules to fix the number of keywords. In our experimentation, we have used long paragraphs consisting of about eight hundred to one thousand words. Therefore, we have taken fixed ten words for each paragraph as per the above generally followed trend.

- For PRK extraction, tokenization based cleansed text generated, i.e., without stop-words, numbers, and duplicate words.
- These words placed in the next column of sentence score summaries.
- The previous PRK methods have drawback regarding the selection of the number of keywords to represent the theme of the document.
- We have resolved this drawback in our proposed technique by selecting the variable number of keywords as per context diversity.

4.5 Variable Length Keywords Creation

We have applied three mainly implemented techniques to generate abstractive and extractive summarizations. Word embedding mainly relates to the document context, whereas the sentence score algorithm generates document specific words frequency counting [11,17]. We have further extracted the TF-IDF based PRK from both extractive forms of summarizations. Then we have taken the intersection of these keywords. The resultant intersection of keywords consists of variable length keywords and effectively handle the diversity for each text paragraph as compared to the fixed-length PRK [22,27]. We have compared all these forms of abstractive and extractive summarization with the actual text. Our proposed technique of abstractive summarization has shown better resemblance to the actual text.

4.6 **Corpus** *Visualization and Single Time Processing*

The proposed *corpus* presentation technique has a one-time cost to fetch the actual paragraph for processing. We have placed the summarized text adjacent to the actual text. Then we have calculated CS of these summaries with the actual text. This technique provides an efficient *corpus* analysis mechanism during text mining [21,25]. However, if there exists some storage issue, there exists an option to generate the *ad hoc* abstractive and extractive summaries. These *ad hoc* summaries have no storage need for further processing. One by one, simultaneous extraction and processing of paragraphs utilizes optimal cost and time for handling the diversity of information in the given text [23,26,32].

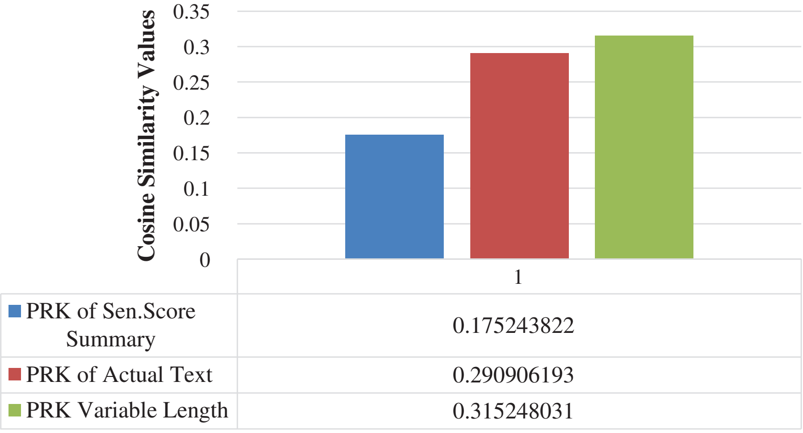
This section contains the detail about the main steps performed in the presented technique. We have discussed the way of expressing the *corpus* with its salient themes. For this, we have processed the extended *corpus* into two stages. At the first stage, the sentence level text cleansing performed, and this text used to extract the main lines of the text. Then we analyzed the text, word by word, and cleansed it to obtain the keyword-based representation of the text. These extracted forms of text substitute lengthy *corpus* text processing and save the overhead of cost and time. We have combined the features of these techniques to generate a robust set of keywords with better similarity to the actual text. We have used Python language packages of NLP to implement the proposed technique.

5 Results and Discussion

We have taken a diverse news dataset [33] for the experimentation. We have selected long paragraphs consisting of more than eight hundred words, see Figs. 2(a) and 2(b). The actual text of the *corpus* has six different reduced forms, represented by multiple columns. In Fig. 2(a), Column H consists of variable length keywords extracted from all forms of summarized text. We have compared these summarizations with the actual text and with each other as shown in Fig. 2(b). We have described these results in this section.

5.1 Average of PRK from Word2Vec Summarized Text Comparison

Word2Vec summarized text comprises of the words which have a higher probability of occurring together in each document [34,35]. We have extracted TF-IDF based PRKs from this text. We have comparted these PRKs with the PRKs of actual text, sentence score, and our proposed Combined PRKs (CPRK). The keywords extracted from this Word2Vec reduced summaries have better average similarity with CPRKs as compared to the remaining two types of abstractive summarizations, see Fig. 3 below.

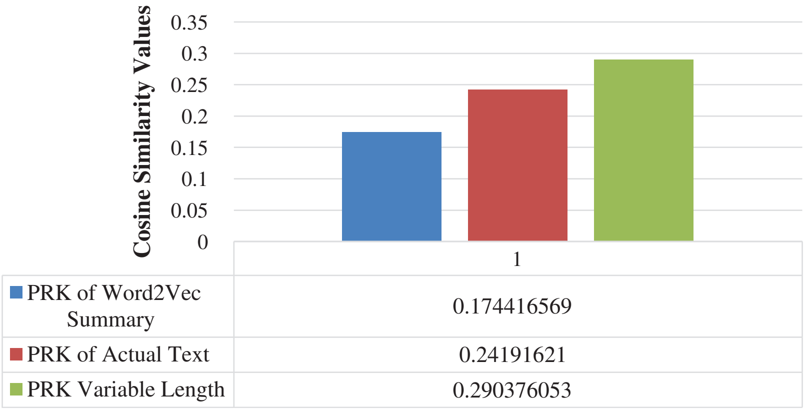


	1
🟦 PRK of Sen.Score Summary	0.175243822
🟥 PRK of Actual Text	0.290906193
🟩 PRK Variable Length	0.315248031

Figure 3: Average of PRK from Word2Vec summarized text comparison, with the PRKs of all three other forms of summarized text

5.2 Average of PRK from Sentence Score Summarized Text Comparison

Sentence score summarized text comprises of the words which have greater frequency to occur in the document. We have extracted TF-IDF based PRKs from this text. We have compared these PRKs with the PRKs of actual text, Word2Vec and proposed CPRKs. The keywords extracted from sentence score summaries have better average similarity with our CPRKs, as compared to the remaining two types of abstractive summarizations as shown in Fig. 4.

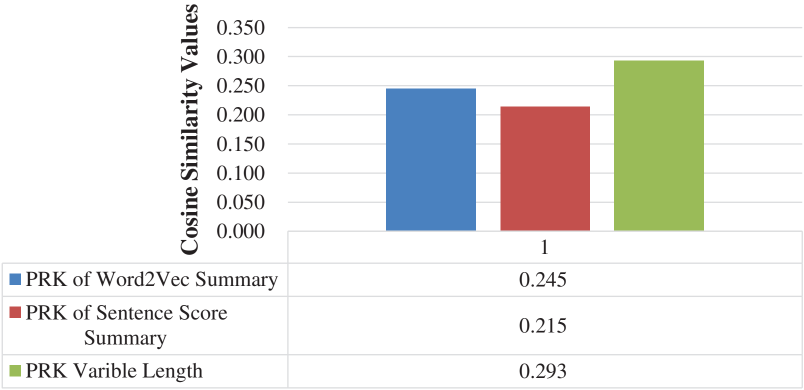


	1
🟦 PRK of Word2Vec Summary	0.174416569
🟥 PRK of Actual Text	0.24191621
🟩 PRK Variable Length	0.290376053

Figure 4: Average of PRK from sentence score summarized text comparison, with the PRKs of all three other forms of summarized text

5.3 Average of PRK from Actual Text Comparison

We have extracted TF-IDF based PRKs from the actual text. We have compared these PRKs with the PRK's of Word2Vec, sentence score and our proposed CPRKs. The keywords extracted from the actual text, have a better average similarity with CPRKs, as compared to the remaining two types of abstractive summarizations as presented in Fig. 5.



	1
🟦 PRK of Word2Vec Summary	0.245
🟥 PRK of Sentence Score Summary	0.215
🟩 PRK Varible Length	0.293

Figure 5: Average of PRK from actual text comparison, with the PRKs of all three other forms of summarized text

5.4 Extractive Text Summarization Comparison

The abstractive summarization from the actual text has better similarity with Word2Vec reduced text, see Fig. 5. The PRKs of Word2Vec, have better similarity with PRK of actual text, as compared to PRKs of sentence score summary, see Figs. 3 and 4. To elaborate on this fact, we have calculated the similarity of extractive summarizations with the actual text. The Word2Vec extractive summaries have better similarity with actual text as shown in Fig. 6.

5.5 Variable Length PRK Extraction

We have taken the intersection of three abstractive summarizations to form a new type of abstractive summarization. This technique provides a very stable variable-length abstractive summarization. It has a smaller number of words as compared to extractive summarizations and gives better text similarity with the actual text. This technique solves the issue to decide that how much keywords are enough to extract from a text paragraph as shown in Fig. 7. We have fixed PRK length as ten. However, this technique has automatically extracted the required number of keywords.

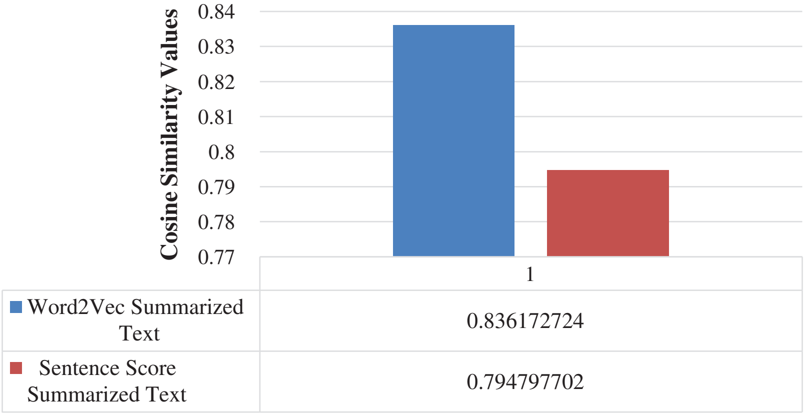


Figure 6: Comparison of extractive summarizations: Word2Vec summary with sentence score summary

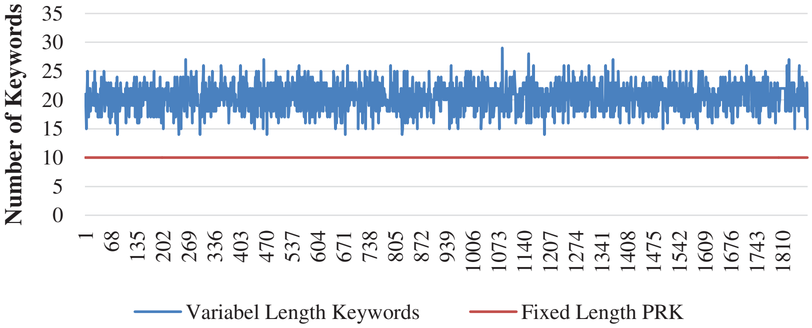


Figure 7: Variable length keywords as per text diversity

This technique has doubled the keywords for actual text paragraphs as shown in Fig. 8. This technique ultimately fulfils the deficiency of the fixed-length TF-IDF based PRKs extractions.

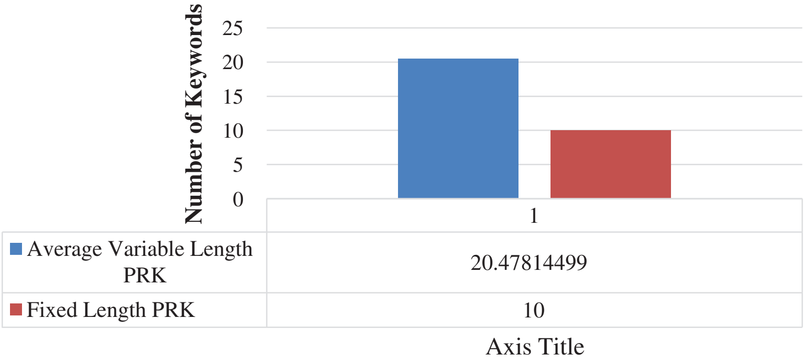


Figure 8: Average of all variable length keywords

All the previous TF-IDF based PRK extraction techniques fail to guarantee the uniform representation of the text documents with extracted keywords. The proposed CPRKs identifies the text paragraphs uniformly as shown in Fig. 9. In this figure, it is evident that CRK has exactly equal similarity with actual text and the remaining two forms of the extractive summarizations.

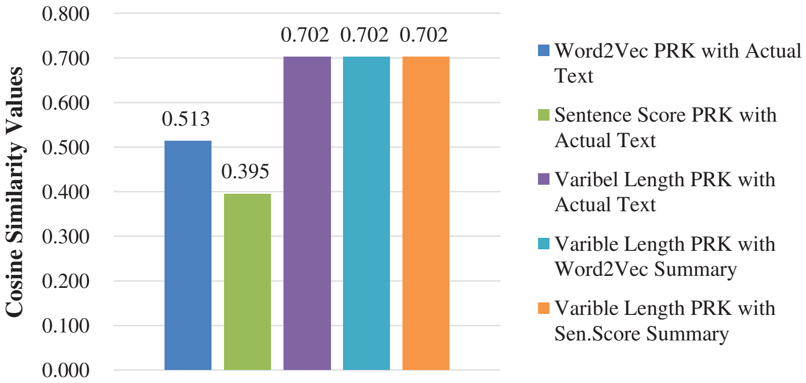


Figure 9: Variable PRK uniformity to identify multiple forms of corpuses

Initially, we have applied *corpus* pre-processing for the removal of text abnormalities. We have created the sentence score summary of each paragraph from the text obtained. We have created a Word2Vec summary of each paragraph. We have applied POS tagging, stemming, lemmatization, and morphological analysis to the original text. We have used this text in our TF-IDF based abstractive summarization. We have created TF-IDF based PRK, from the cleansed text. We have placed these summaries in adjacent columns of our CSV *corpus* file. For all the above summarizations, we have calculated the cosine similarities with the actual text. These values provided a comparative analysis of all the text summarization. We have further extracted TF-IDF based PRK from both types of extractive summarizations, obtained by SSA & GW2VA. We have combined the keywords extracted from extractive summarization. Above provided a unique set of keywords, as variable-length abstractive summarization, for each paragraph. The CS values comparisons for each reduced *corpus*, presented in the form of table & graphs, to show the efficiency of our applied technique.

6 Conclusion

The existing studies show that we cannot justify text processing efficiency, just in terms of reduced time & cost. This assessment fundamentally begins with justifying the robustness of the applied metadata generating technique. After that, reduced processing cost becomes justifiable if the metadata sufficiently substitutes the actual text. All this creates the need for designing optimal metadata techniques. These techniques have an essential role in the system of text information retrieval. This technique is different from other TF-IDF based PRK extractions. The previous techniques overlook the information diversity of the multiple text documents. The fixed-length keywords do not provide uniform text identification for every text paragraph. The current technique implements the principles of deep learning and interrelates the number of keywords of a paragraph with the diversity of its information. Therefore, every paragraph has an equally capable set of keywords to present the actual theme of the paragraph. In this paper, we have created three different summarized representations of the actual text. The first form of the summarization is TF-IDF based PageRank Keywords. The second and third are sentence score & Gensim-Word2Vec Extractive summaries of the text. We have calculated the Cosine Similarity of these reduced forms with the actual text *corpus*. We have presented a detailed comparative analysis of these techniques. Based on these summarizations, we have suggested our improved technique of abstractive summarization with variable length PRKs.

Acknowledgement: We are thankful to Government College University Faisalabad (GCUF), Pakistan to provide resources for this research.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

[1] S. Qaiser and R. Ali, “Text mining: Use of TF-IDF to examine the relevance of words to documents,” *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25–29, 2018.

[2] R. K. Roul, J. K. Sahoo and K. Arora, “Query-optimized PageRank: A novel approach,” in *Proc. Computational Intelligence in Data Mining*, Singapore: Springer, pp. 673–683, 2019.

[3] S. Ganiger and K. M. M. Rajashekharaiah, “Comparative study on keyword extraction algorithms for single extractive document,” in *Proc. Second Int. Conf. on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India: IEEE, pp. 1284–1287, 2018.

[4] Y. R. Gu and M. X. Xu, “Keyword extraction from News articles based on PageRank algorithm,” *Journal of the University of Electronic Science and Technology*, vol. 46, no. 05, pp. 777–783, 2017.

[5] S. Pan, Z. Li and J. Dai, “An improved TextRank keywords extraction algorithm,” in *Proc. ACM Turing Celebration Conf.*, China: ACM, pp. 1–7, 2017.

[6] W. Li and J. Zhao, “TextRank algorithm by exploiting Wikipedia for short text keywords extraction,” in *Proc. 3rd Int. Conf. on Information Science and Control Engineering (ICISCE)*, Beijing, China: IEEE, pp. 683–686, 2019.

[7] D. Mahata, R. R. Shah, J. Kuriakose, R. Zimmermann and J. R. Talburt, “Theme-weighted ranking of keywords from text documents using phrase embeddings,” in *Proc. 2018 IEEE Conf. on Multimedia Information Processing and Retrieval (MIPR)*, Miami, FL: IEEE, pp. 184–189, 2018.

[8] W. Zheng, S. Mo, P. Duan and X. Jin, “An improved PageRank algorithm based on fuzzy C-means clustering and information entropy,” in *Proc. 3rd IEEE Int. Conf. on Control Science and Systems Engineering (ICCSSE)*, Beijing: IEEE, pp. 615–618, 2017.

[9] C. Fang, D. Mu, Z. Deng and Z. Wu, “Word-sentence co-ranking for automatic extractive text summarization,” *Expert Systems with Applications*, vol. 72, pp. 189–195, 2017.

[10] Z. Nasar, S. W. Jaffry and M. K. Malik, “Textual keyword extraction and summarization: State-of-the-art,” *Information Processing & Management*, vol. 56, no. 6, 102088, 2019.

[11] C. F. Tsai, K. Chen, Y. H. Hu and W. K. Chen, “Improving text summarization of online hotel reviews with review helpfulness and sentiment,” *Tourism Management*, vol. 80, pp. 104–122, 2020.

[12] Q. Aziz, I. A. Farha, W. G. M. Washaha and E. Maali, “An efficient single document Arabic text summarization using a combination of statistical and semantic features,” *Journal of King Saud University-Computer and Information Sciences*, 2019 (In press).

[13] L. Oneto, F. Bisio, E. Cambria and D. Anguita, “Statistical learning theory and ELM for big social data analysis,” *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 45–55, 2016.

[14] M. Akbari, X. Hu, F. Wang and T. S. Chua, “Wellness representation of users in social media: Towards joint modelling of heterogeneity and temporality,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2360–2373, 2017.

[15] T. Lebo, N. Del Rio, P. Fisher and C. Salisbury, “A five-star rating scheme to assess application seamlessness,” *Semantic Web*, vol. 8, no. 1, pp. 43–63, 2017.

[16] N. Gina, A. Tanweer, B. Fiore-Gartland and L. Osburn, “Critique and contribute: A practice-based framework for improving critical data studies and Data Science,” *Big Data*, vol. 5, no. 2, pp. 85–97, 2017.

[17] D. Gkatzia, O. Lemon and V. Rieser, “Data-to-text generation improves decision-making under uncertainty,” *IEEE Computational Intelligence Magazine*, vol. 12, no. 3, pp. 10–17, 2017.

[18] J. Androutsopoulos, “Languaging when contexts collapse: Audience design in social networking,” *Discourse Context & Media*, vol. 4, pp. 62–73, 2014.

[19] B. Monika and H. Caple, “‘Value added’: Language, image and news values,” *Discourse Context & Media*, vol. 1, no. 2–3, pp. 103–113, 2012.

[20] S. Poria, E. Cambria, A. Gelbukh and F. Bisio, “Sentiment data flow analysis by means of dynamic linguistic patterns,” *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 26–36, 2015.

[21] H. Heand and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[22] K. Ravi and R. Vadlamani, “A survey on opinion mining and sentiment analysis,” *Knowledge-Based Systems*, vol. 89, pp. 14–46, 2015.

[23] R. Dale, “The pros and cons of listening devices,” *Natural Language Engineering*, vol. 23, no. 6, pp. 969–973, 2017.

[24] J. M. Luzón, “This is an erroneous argument: Conflict in academic blog discussions,” *Discourse Context & Media*, vol. 2, no. 2, pp. 111–119, 2013.

[25] A. Bagnall, J. Lines, J. Hills and A. Bostan, “Time-series classification with COTE: The collective of transformation-based ensembles,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522–2535, 2015.

[26] M. Lippi, “Statistical relational learning for game theory,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 8, no. 4, pp. 412–425, 2016.

[27] J. J. Li, H. Yang and H. Tang, “Feature mining and sentiment orientation analysis on product review,” in *Proc. Management Information and Optoelectronic Engineering: Proc. of the 2016 Int. Conf. on Management, Information and Communication (ICMIC2016) and the 2016 Int. Conf. on Optics and Electronics Engineering (ICOEE2016)*, Singapore, pp. 79–84, 2017.

[28] V. Buskens, C. Snijders and C. Snijders, “Effects of network characteristics on reaching the payoff-dominant equilibrium in coordination games: A simulation study,” *Dynamic Games and Applications*, vol. 6, no. 4, pp. 477–494, 2016.

[29] M. Dojchinovski and T. Vitvar, “Linked web APIs dataset,” *Semantic Web Preprint*, vol. 9, no. 4, pp. 1–11, 2016.

[30] F. Pallavicini, P. Cipresso and F. Mantovani, “Beyond sentiment: How social network analytics can enhance opinion mining and sentiment analysis,” *Sentiment Analysis in Social Networks*, pp. 13–29, 2017.

[31] Y. Ali, A. G. Shahraki and O. R. Zaiane, “Current state of text sentiment analysis from opinion to emotion mining,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–33, 2017.

[32] A. Viswam and G. Darsan, “An efficient bitcoin fraud detection in social media networks,” in *Proc. Int. Conf. on Circuit Power and Computing Technologies (ICCPCT), 2017*, Kollam, India, IEEE, pp. 1–4, 2017.

[33] Kaggle. https://www.kaggle.com/sunnysai12345/news-summary.

[34] Y. Liu, J. Wang, Y. Jiang, J. Sun and J. Shang, “Identifying the impact of intrinsic factors on topic preferences in online social media: A nonparametric hierarchical Bayesian approach,” *Information Sciences*, vol. 423, pp. 219–234, 2018.

[35] C. P. Chen and C. Y. Zhang, “Data-intensive applications, challenges, techniques and technologies: A survey on Big Data,” *Information Sciences*, vol. 275, pp. 314–347, 2014.