Information Processing and Management 58 (2021) 102536

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Text summarization using topic-based vector space model and semantic measure

Ramesh Chandra Belwal *, Sawan Rai, Atul Gupta

Department of Computer Science and Engineering, Indian Institute of Information Technology, Design and Manufacturing, Jabalpur, India

ARTICLE INFO

Keywords:
Extractive summarization
Topic modeling
Relevance measure
Vector space model
Semantic measure

ABSTRACT

The primary shortcoming associated with extractive text summarization is redundancy, where more than one sentence representing a similar type of information are incorporated in summary. In the last two decades, a lot of extractive text summarization methods have been proposed, but less attention was paid to the redundancy issue. In this paper, we propose a text summarization technique that incorporates topic modeling and semantic measure within the vector space model to find the extractive summary of the given text. Our main objective is to address the redundancy problem associated with summarization methods and include only those sentences in summary, which represent the maximum of the topics embedded in the given text document. We generate the topic vector of the given document by representing the sentences in an intermediate form using a vector space model and topic modeling. Moreover, to make the proposed method efficient, we incorporate the semantic similarity measure to find the relevance of the sentence. We introduce two different ways to create the topic vector from the given document, i.e., Combined topic vector and Individual topic vector approach. Evaluation results on two datasets show that the summaries generated by both variants (Combined and Individual topic vector techniques) of the proposed method are found to be closer to the human-generated summaries when compared with the existing text summarization methods.

1. Introduction

With the excessive overloading of data on the Internet, the users are facing difficulty to find relevant information in the desired time. It is difficult for human beings to manually summarize the large text documents in the efficient manner. Nobody has enough time to go through it all, and sometimes we have to make important choice based on what we can assimilate (Maybury, 1999). As the amount of information increasing continuously, the systems that can automatically summarize the input documents are desirable (Radev, Hovy, & McKeown, 2002).

Text summarization is the process of converting the text document in a shorter form which helps users to save their time and effort to have the gist of the original documents. The text summary contains the major points of the document.

Text summarization can be achieved in fundamentally two different ways, i.e., Extractive and Abstractive (Gupta & Lehal, 2010). The extractive system extracts a few sentences from the entire collection without modifying those sentences. On the other hand, abstractive summarizers may even produce new sentences to the summary or modify the original sentences. These types of summarizers produce important sentences like human-generated summaries.

Text summarization can be either query-based or generic. In query-based summarization, the goal is to generate a summary that is relevant to the given query (Abdi, Shamsuddin, & Aliguliyev, 2018; Van Lierde & Chow, 2019; Yousefi-Azar & Hamey, 2017). We

* Corresponding author.
E-mail addresses: rameshbelwal@gmail.com (R.C. Belwal), sawanrai@iiitdmj.ac.in (S. Rai), atul@iiitdmj.ac.in (A. Gupta).

https://doi.org/10.1016/j.ipm.2021.102536
Received 22 April 2020; Received in revised form 9 November 2020; Accepted 24 January 2021
Available online 9 February 2021

propose a generic summarization technique where a generalized summary is generated independently of any specific information or query.

Extractive text summarizers select the most relevant sentences from the given input text, which can be within a single document or a group of documents (Nenkova & McKeown, 2012). Generally, the information content in the given document is not equally scattered in each sentence. Therefore, it would be efficient to find the subset of sentences, which serves as the summary of the document (Radev et al., 2002).

Text summarization was primarily begun with feature-based techniques, where various linguistic and statistical features have been used to find the relevance of the sentences to be included in summary. Various statistical and linguistic features have been proposed to be used as attributes for finding the relevance and similarity among the sentences (Abdi, Shamsuddin, Hasan, & Piran, 2018; Fattah & Ren, 2008; Gupta, Pendluri, & Vats, 2011; Lloret & Palomar, 2009; Mutlu, Sezer, & Akcayol, 2019).

Relevance measure (RM) based methods use statistical similarity measures to find the relevance of the sentences to be included in the summary (Gong & Liu, 2001). Topic-based techniques are based on topic word distribution in the input document and modeling techniques have been incorporated in the summarization method to find the summary (Blei, Ng, & Jordan, 2003; Chang & Chien, 2009; Hu, Sun, & Lim, 2008; Na et al., 2014; Nguyen, Tran, Nguyen, & Nguyen, 2019).

In LSA based techniques, the singular value decomposition has been used to reduce the dimension of the sentence vector. Gong and Liu (2001) and Ozsoy, Alpaslan, and Cicekli (2011) have made the use of LSA for text summarization in their work. Haiduc, Aponte, Moreno, and Marcus (2010) have used VSM and LSI based methods to find a summary of source code documents. The Graph-based ranking algorithm in which knowledge drawn from the input text is used to rank the sentences to the summary (Barrios, López, Argerich, & Wachenchauzer, 2015; Mihalcea & Tarau, 2004).

Recently machine learning techniques (RNN, CNN, LSTM, Encoder–Decoder based) are extensively used for text summarization where trainable summarizers are used to find the parameters for the algorithms (Fu, Wang, Zhang, Wei, & Yang, 2020; Fuad, Nayeem, Mahmud, & Chali, 2019; Mao, Yang, Huang, Liu, & Li, 2019; Nallapati, Zhai, & Zhou, 2017; Narayan, Cohen, & Lapata, 2018; Zhang, Lapata, Wei, & Zhou, 2018).

In practice, each statement of any given text represents a topic embedded in the document. Text summarization methods select those sentences that either describe a topic or very close to the overall document. In this way, many times, either lengthy or the sentences with redundant information are included in the summary. Along with this, sometimes perfect sentences are missed to be included in a summary because of not considering the semantic meaning of words.

We propose an unsupervised method of extractive summarization, that selects only those sentences containing a maximum of the topic embedded in the given document. We have designed our method in such a way, topic vector generation and relevance finding steps are autonomous so that it remains flexible and adaptable. The summary generated by the proposed approach includes only those sentences that represent the essential topic words even if the sentence is short in length, unlike involving lengthy sentences as in RM based methods.

The proposed method uses topic modeling to find the relevance (or importance) of the sentences to be included in the summary. Topic modeling is a technique that identifies words describing the topics of the input document (Blei et al., 2003). The output generated by topic modeling is the topics and respective words associated with the topics. It also gives the probability of words of the respective topics. We use the words with high probabilities as the representative of the topics and, the relevance of the sentence is calculated based on the similarity with the topic words. To further improve the similarity measure, we incorporate the semantic meaning to find the similarity between each sentence and topic vector.

We have introduced two variations of the proposed method, i.e., Combined vector approach and Individual vector approach. In Individual vector approach, each sentence represents a single topic. On the other hand, the Combined vector approach includes those sentences in summary, which contain words from the multiple topics embedded in the input document.

In the proposed method, the sentences of the input document are represented in vector form using VSM (Vector Space Model) (Salton, Wong, & Yang, 1975). The Vector Space Model represents each sentence of the input document D as an n-dimensional vector where n is the total number of distinct terms used in the document. The proposed method uses three major steps for generating the summary. The first step is to generate the topic vector of the given document (i.e., topic word representation of the input document). Subsequently, we find the relevance of each sentence on the basis of similarity with the topic vector. The last step of the proposed method is the ranking of the sentences on the basis of the relevance so that top-ranked sentences are included in the summary.

As we know, the statistical techniques used in most of the relevance based methods cannot deal with synonyms, hypernyms, hyponyms, etc. To overcome this problem, we incorporate semantic similarity measure in topic modeling to find the rank of the candidate sentences for the summary. The procedure presented in the proposed method keeps the topic generation, and relevance creation steps independent of each other to make the algorithm more flexible and adaptable. The process selects only those sentences in summary, which contain maximum topics within them.

Automatic evaluation on CNN/DailyMail (Hermann et al., 2015) and Opinosis (Ganesan, Zhai, & Han, 2010) datasets show that the summary generated by the proposed method scores better ROUGE values than the other text summarization methods.

In addition to automated evaluation, the manual assessment has shown that the proposed approach has substantial improvement over current state-of-the-art summarization methods. For the manual evaluation, the summary is evaluated on the criteria of informativity and coherence.

Another positive aspect of the proposed approach is that it can work efficiently for any language other than English, provided that the following two requirements are fulfilled (1) There needs to be a database that can assist the topic modeling for the given language. The database should support stop words, stemming and lemmatization, etc. The availability of databases is not even

a major concern because multilingual topic modeling tools are available nowadays. (2) Like WordNet, a database should exist in which the semantic meaning of that language's words is available. If we do not have this database available, the proposed technique will still be incorporated for the given language. We have designed similarity calculation and conceptual steps of vector generation so that we can also incorporate another approach (like statistical approaches) in the similarity calculation instead of the semantic approach.

The rest of the paper is organized as follows: Related work is described in Section 3. The proposed method is elaborated in Section 4. Section 5 demonstrates the evaluation results. Conclusion and future work is presented in Section 6.

2. Research objective

The importance of sentence in relevance measure based method depends on how much the particular sentence is statistically closer to the overall document. As a result, most of the time, long sentences are included in the generated summary. In the topic-based method, each sentence assumed to represent one of the topics embedded in the input document. In this way, many times, more than one sentence included in the summary, represents a similar topic. The primary constraint with the learning-based method is the prerequisite of both input documents and respective summaries to learn the parameter for the algorithms. As statical feature-based methods do not consider the semantic meaning of words, many times, good sentences are ignored to be included in the summary. Whereas, most of the abstractive methods are complex and require linguistic knowledge for the implementation.

Considering the above-said shortcomings, we attempt to address the following objective in the proposed work.

1. If a sentence is relatively short but does contain many topics, then how to select it to be included in the summary.
2. How to overcome redundancy problem associated with the summarization system.
3. How to incorporate semantic meaning while finding the relevance of the sentences.
4. How to design the algorithm in such a manner that makes it flexible and adaptable for future changes.
5. How the appropriate responses of the above questions are leveraged with the efficient summary of the given text.

3. Related work

Extractive summarization methods produce text summaries by choosing the most significant sentences from the original document. The importance of the sentence is determined on the basis of linguistic and statistical features of the given text. In every text document, there are a couple of sentences that are although short in length but cover the maximum topics embedded in the overall document. The proposed technique targets those sentences to be included in the summary.

The proposed work is an unsupervised text summarization method, which incorporates the topic modeling in the Vector Space Model to find the extractive summary of the given document. An important step of any extractive text summarization technique is to find the features or parameters of the input text so that the best sentences for the summary can be identified. In a feature-based approach, statistical and linguistic features are used to score the sentences so that suitable candidate sentences can be identified for the summary.

Ferreira et al. (2013) performed a detailed analysis and assessment of fifteen algorithms for the sentence scoring. Lloret and Palomar (2009) proposed an approach combining three different features (word frequency, textual entailment, and the code quantity principle) to produce extracts from newswire documents. Mani and Bloedorn (1998) made the use of machine learning techniques on the corpus of documents and respective abstracts to discover the best combination of features for the summarization. Abdi, Shamsuddin, Hasan, and Piran (2018) used several features into a unified set to develop an accurate classification system for the summarization. Mutlu et al. (2019) presented a study on different hand-crafted features used for text summarization. Finding the best set of features for the text summarization techniques is still a challenging task.

The proposed method is centered around the relevance measure based extractive text summarization. These are the information retrieval based techniques where sentences can be represented as vector form (i.e. term-document matrix), and different similarity measures are used to find the similarity among the sentences. Vector Space Model (VSM) (Salton et al., 1975) and Latent Semantic Indexing (LSI) (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) are widely used techniques for text summarization. Gong and Liu (2001) presented the technique that extracts the sentence from the input document and on the basis of the ranking of the sentences the summary has been found. They used IR (information retrieval) and LSA (Latent Semantic Analysis) in their work. IR methods are used for generating the sentence relevance and respective rank. The highest K ranked sentences are included in the summary. Where relevance of the sentence is decided on the basis of its similarity with the overall document. The term frequency vectors of both the sentences and the overall document are generated, and the inner product of the vectors yields the relevance of the sentence. The relevance based methods use the statistical techniques to find the relevance of the sentences to be included in the summary. In our method, we incorporate the semantic meaning of the sentence while finding the similarity among the sentences. The inclusion of the semantic meaning improves over the performance while finding the similarity among topic words and each sentence.

However, the proposed method is based upon the original sentence vectors to find the relevance of the sentences, but there are other techniques like LSA (Deerwester et al., 1990) that work on reduced dimension vector for the summarization task. In LSA (Latent Semantic Analysis), the concept of singular value decomposition (SVD) is used where the dimension of the document vector can be reduced while similarity is preserved. LSA is a technique for investigating the relationships between the documents and the terms they contain. The concept of singular value decomposition is used to reduce the number of rows, although preserving the similarity structure among the columns. Gong and Liu (2001) and Ozsoy et al. (2011) have made the use of LSA for text summarization in their

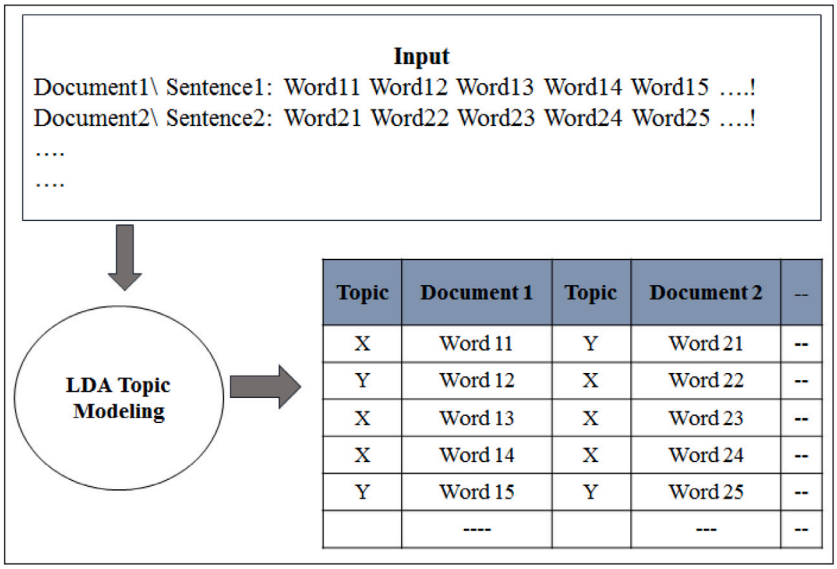

Fig. 1. LDA approach.

work. Haiduc et al. (2010) have used VSM and LSA based methods to find a summary of source code documents. These methods do not incorporate the semantic meaning while finding the relevance of the candidate sentences for the summary. Yeh, Ke, Yang, and Meng (2005) have used the trainable summarizer and LSA to find the summary of the document.

Our technique uses topic modeling to find the extractive summary of the given document. Topic modeling is the technique that is used to identify the words which describe the topic of the given document. Some of the most common summarization methods rely on topic representations, and this class of approaches shows a remarkable variety in sophistication and power of presentation (Nenkova & McKeown, 2012).

Blei et al. (2003) presented a Latent Dirichlet Allocation (LDA) topic modeling technique. LDA is a generative probabilistic model that is applied on discrete data collection, such as text corpus.

The model assumes each word in the document as an attribute to one of the document's topics. The basic idea behind LDA is that each of the documents is represented as random mixtures over latent topics, and each topic is characterized by a distribution over words. After the iterative calculation of the topic and word probabilities, LDA generates the output as Fig. 1. Details of how the topic modeling is incorporated to generate the topic vector from the given text document are elaborated in the next Section 4.

Chang and Chien (2009) proposed the sentence-based Latent Dirichlet Allocation for document summarization. Hu et al. (2008) introduced the summarization technique, which produces the blog data summary using the topics discussed among the reader's comments. Na et al. (2014) proposed a method that models content and respective titles, mixing them by learning methods. Topic modeling is also used in text analytic tasks, such as, Lim, Buntine, Chen, and Du (2016) have used Bayesian topic modeling in social media to model the text data. Cuong, Tran, Van, and Than (2019) have investigated the various strategies to incorporate dropout in topic models to avoid the over-fitting for short text. Amplayo and Song (2017) have proposed fine-grained sentiment analysis for the summarization of the short texts using topic modeling. Zhang, Wu, Bu, Jiang, and Cao (2018) have introduced a pattern-based topic detection to be applied on tweets and summarize them. Barros, Lloret, Saquete, and Navarro-Colorado (2019) have introduced an abstractive summarizer (named NATSUM) to generate the summary from various news documents related to the same topic. Li et al. (2018) have proposed a topic model, CSTM (common semantics topic model) by introducing common topic, to gather the noisy words.

In the proposed method we use the topic modeling to find the theme of the overall document and apply the semantic measure to find the closeness of the sentence with the topic theme of the input document. To make it flexible for the similarity measure, we have kept two steps (i.e., topic representation and sentence ranking) independent of each other. In our work, we do not need the titles of the given documents.

VSM (Vector Space Model) based methods create a term-document matrix to represent the sentences. However, there are different techniques like graph-based approaches that create a graph for the input text, and ranking algorithms are utilized for text summarization. In a graph-based approach, sentences are represented as nodes of the graph, and graph-based ranking methods (i.e. PageRank, Page, Brin, Motwani, & Winograd, 1999, HITS, Kleinberg, 1999, etc.) are applied to find the rank of the nodes (sentences). The final summary is generated on the basis of the rank of the sentences. Mihalcea and Tarau (2004) introduced a graph-based ranking algorithm called TextRank, in which knowledge drawn from the input text is used for ranking or selection decisions. To improve over the original TextRank algorithm, Barrios et al. (2015) proposed the alternative of the similarity measure.

In graph-based text summarization, the graph is first created using the sentences given in the input text. Then the weight is assigned among the edges using various similarity measures. In TextRank, Mihalcea and Tarau (2004) used the following similarity measure to assign the weight to edges (i.e., edges A and B) of the graph

$$\text{Similarity}(A, B) = \frac{|W_k|W_{k\in A}\&W_{k\in B}|}{\log(|A|)+\log(|B|)}$$

But in the above method, the semantic meaning has not been incorporated. Consequently, sometimes, even though more than one sentence containing different words expresses the same sense, the edge is assigned a zero weight. There is also another problem with graph-based methods that every sentence must measure the similarity value together with each other sentence while assigning the weight to the edges. The process must be repeated for every node. On the other hand, after the topic vector is generated in the proposed method, each sentence is compared with the topic vector only once.

The recent success of machine learning techniques in text analysis field (Chen, Cai, Chen, & de Rijke, 2019; Liu & Jansen, 2017; Ma et al., 2016; Salminen et al., 2020) played a significant role in the field of text summarization also. The summarization techniques can use either a supervised or unsupervised approach. The supervised techniques use the human-generated summaries to find the features or parameters of summarization algorithms (Fattah & Ren, 2009; Oyedotun & Khashman, 2016; Song, Huang, & Ruan, 2019). Whereas unsupervised techniques determine the features or parameters without using human-made summaries (Alguliev, Aliguliyev, Hajirahimova, & Mehdiyev, 2011; Barzilay & McKeown, 2005; Gong & Liu, 2001; Harabagiu, Lacatusu, & Morarescu, 2002; Mihalcea & Tarau, 2004).

Extensively used supervised methods for text summarization are mainly based on RNN, CNN, LSTM, Encoder–Decoder techniques. Nallapati et al. (2017) introduced SummaRuNNer, an RNN based model for extractive summarization. They (Nallapati et al., 2017) used an unsupervised method for converting the abstractive summaries to extractive labels.

Narayan et al. (2018) have used reinforcement learning for extractive text summarization. Zhang, Lapata, Wei, and Zhou (2018) proposed a latent variable extractive technique by leveraging human-generated summaries. They (Zhang, Lapata, Wei, & Zhou, 2018) used the sentence compression model for summarization. Fuad et al. (2019)used neural sentence fusion and sentence clustering for the multi-document summarization. Mao et al. (2019) proposed the extractive summarization method that combines supervised and unsupervised learning techniques. The primary constraint associated with the learning-based techniques is the need for a large amount of instance to be learned. Whereas, our method can work in a small number of instances very well.

The proposed method is an unsupervised text summarization technique that incorporates the use of topic modeling in the relevance measure based concept. From the available literature, it can be observed that the shortcomings associated with the extractive methods are mainly redundancy and lack of semantic measure for sentence selection. As it is observed, the words of the input text usually are associated with the topics embedded in the text itself. To overcome the redundancy problem, we have incorporated the topic modeling in such a manner that it can select the sentences containing maximum topic words. Moreover, we have also included the semantic measures while finding the ranks of the sentences to be included in the summary. In the proposed method, we use the concept of LDA topic modeling to find the relevance of the sentence to be included in the summary. The detailed steps of the procedure used in the proposed work are described in the next section.

4. Proposed method

Text summarization system proposed in this paper is based on the similarity of sentences with the topic word embedded in the input text. Usually, human beings have a tendency to write some of the sentences in the text which cover overall topic themes of the given document. The proposed technique tries to include those sentences in the summary which are closer (or similar) to the topic words of the given document.

In the proposed method, we use the topic modeling to create the topic vector (denoted as T_v). Topic vector is a vector which consists of a subset of words (W words of P topics) of the input document. We compute the similarity score between each sentence S_i of the input document and the topic vector T_v. The importance (or relevance) of the particular sentence is calculated based on how much the sentence is similar to the topic vector.

The process of generating topic vector T_v and finding the relevance of the sentences can be performed in multiple ways, which are described later in this section. We use LDA topic modeling (Blei et al., 2003) to generate the topics and respective words embedded in the topics. The output generated by the topic modeling approach is the set of words embedded in the topics representing the original document. We want to generate the summary in such a way that it can cover the maximum number of the topics of the input document. Fig. 2 shows the general steps followed in the proposed methodology. Although LDA is introduced for document modeling, it works on sentence level as well. The straightforward method it to treat the individual sentences as the document.

Automatic text summarization is a complex process that needs to be divided into modules (Torres-Moreno, 2014). Therefore to reduce the complexity of the proposed method, we split the overall process into separate parts. The proposed method uses the following steps to generate the extractive summary of the input text.

4.1. Preprocessing

The first step of the proposed method is to preprocess the input text document. Initially, the input document is divided into a set of sentences, followed by text cleaning. We know that in its basic form, the text is the set of tokens, not annotated with the properties (Aggarwal, 2018). We would like to remove those tokens which have little meaning in the context of the input document, i.e., stop words, punctuation, etc. In our work, we use stop word removal, punctuation removal and lemmatization. The goal of both lemmatization and stemming is to convert a word to a common base form. For example, ‘learning’ and ‘learns’ should be changed to its base form as ‘learn’.

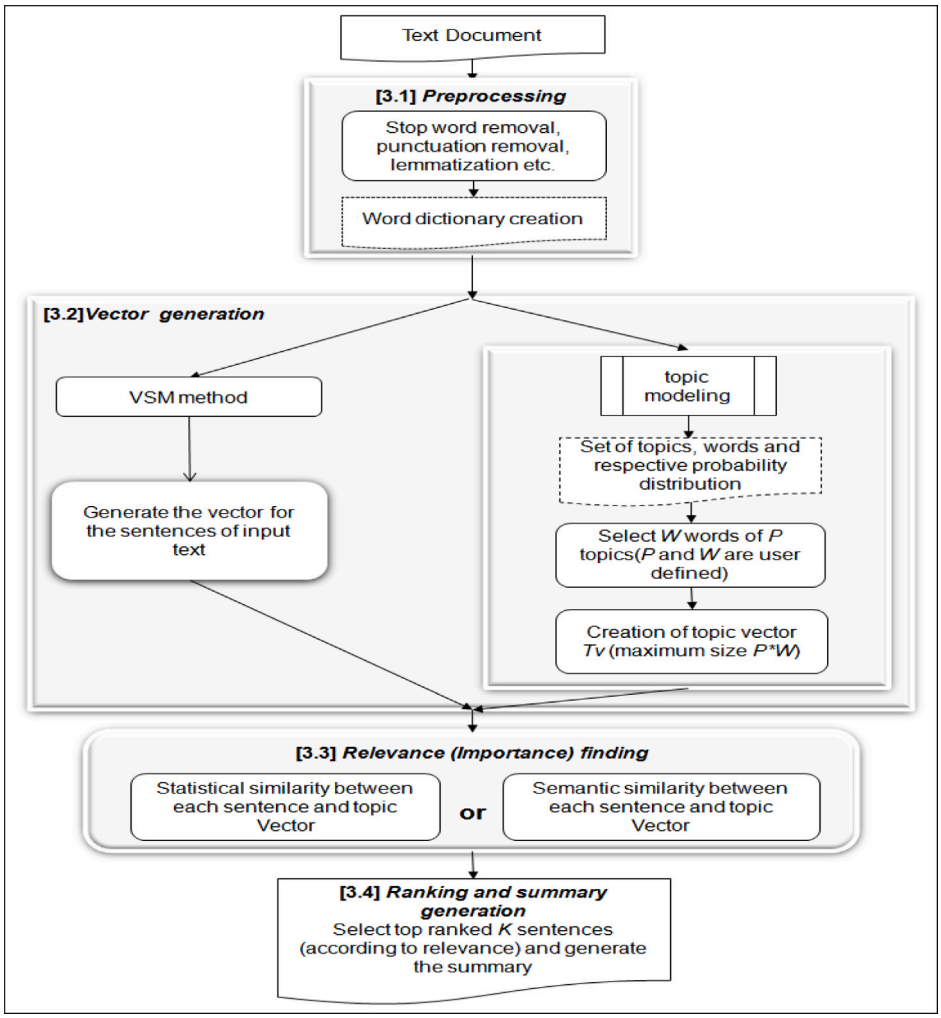

Fig. 2. Steps followed by the proposed method.

4.2. Vector generation

The second step of the proposed method is to represent the input text and topic words in vector form. At first, each of the input sentences is represented in the vector form. Binary and tf-idf are the frequently used representation of the sentences in the vector form. After that, a separate vector is generated with the help of topic modeling methodology. Here we use LDA methodology to generate the topic vector. Top W words based on the probability associated with the words are selected from P topics, where W and P are to be chosen by the user. LDA (Latent Dirichlet Allocation) is a type of topic modeling algorithm used to learn the representation of the topics and topic distribution (how the words of the documents are distributed with the topic) in a given document (Blei et al., 2003). LDA learns topic representation in the following manner

1. First, the number of topics which are to be discovered is selected.

2. Thereafter, LDA (Latent Dirichlet Allocation) checks through each of the words in each of the documents, and randomly assigns the word to one of the topics so that we have documents represented in terms of topics. This random assignment already gives us both topic representations of all the documents and word distributions of all the topics (although it is not very good).

3. Now LDA checks all the documents, the percentage of times that particular word has been associated with a particular topic. LDA then calculate

$p(T|D)$ = percentage of words in a D that are currently assigned to T.(where D is the document and T is the number of topics to be assigned to document) and

$p(W|T)$ = percentage of times the word W was assigned to T overall documents.

4. Reassign W to a new topic, on the basis on value $p(T|D) * p(W|T)$

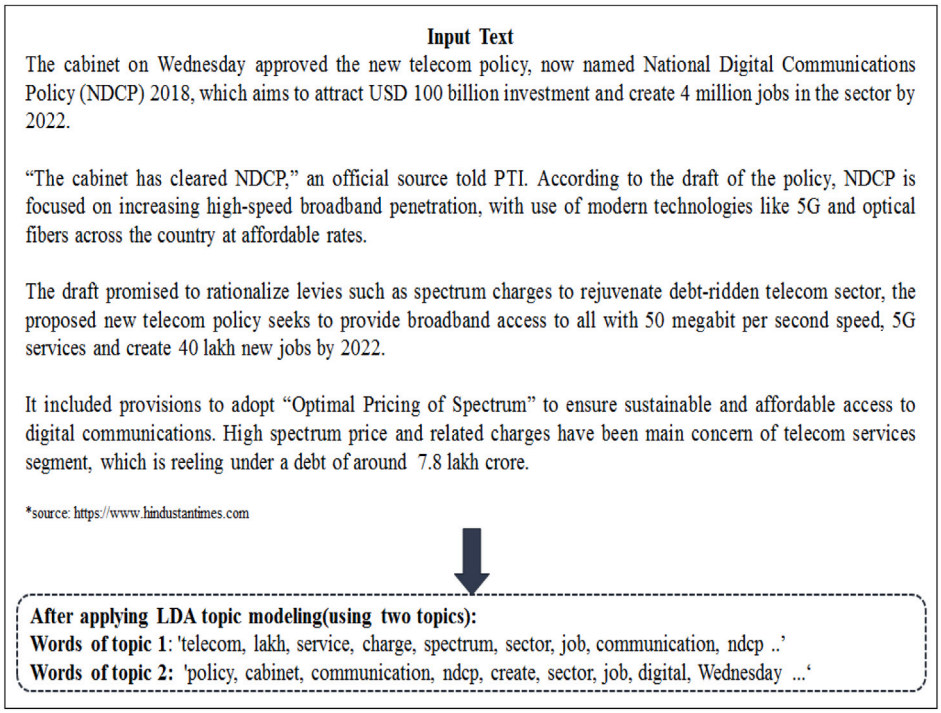

Fig. 3. LDA example (Input text and respective topic words).

5. The above procedure of topic assignment is repeated for each word in every document, iteratively till convergence. The output generated by LDA modeling is the topics and respective words associated with the topics. It also gives the probability of words associated with the topics.

4.3. Relevance (or importance) finding

The next step of the proposed approach is to find the similarity of each of the sentence vector with the topic vector. Commonly used approaches to find the similarity between the different text data are Cosine similarity, Jaccard coefficient, Euclidean distance, etc. Any of the following similarity measures can be used to find the similarity between the sentences of the text document and topic vector generated so far.

Cosine Similarity: The cosine similarity between sentences, A and B can be calculated as

$$Cosine\ Similarity = \frac{\sum_{i=1}^{N} A_i B_i}{\sqrt{\sum_{i=1}^{N} A_i^2}\sqrt{\sum_{i=1}^{N} B_i^2}}$$ (where A_i and B_i are the components of vector A and B respectively).

Jaccard Coefficient: The jaccard coefficient between sentences, A and B can be defined as

$$Jaccard\ \ Coefficient = \frac{|W_A \cap W_B|}{|W_A \cup W_B|}$$ (where W_A and W_B are the words in sentence A and B respectively)

Euclidean Distance: The euclidean distance between sentences, A and B can be defined as

$$Euclidean\ \ Distance = \sqrt{\sum_{i=1}^{N} |A_i - B_i|^2}$$

We have incorporated semantic meaning while finding the similarity between each sentence and topic vector.

Semantic similarity: Between sentence, A and B is computed on the basis of semantic similarity between each of the words in A, to each of the words in B with the help of distance measure based on WordNet. WordNet groups English words into sets of synonyms by recording the relations among the synsets (Miller, 1995). Li, McLean, Bandar, Crockett, et al. (2006) proposed a methodology to find the semantic similarity among the sentences based on structured lexical database and corpus statistics.

In addition to the above methods for finding the statistical or semantic similarity among the sentences, the graph-based extractive text summarizers introduced different similarity measures for the sentences. Mihalcea and Tarau (2004) used the following measure to find the similarity between the sentence A and B

$$\text{Similarity}(A, B) = \frac{|W_k|W_{k\in A} \& W_{k\in B}|}{\log(|A|)+\log(|B|)}$$

Apart from all these, we can use other learning-based techniques, i.e., Word2vec/Doc2vec (Le & Mikolov, 2014; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), to implement the proposed method. We have to pass the sentence (of the input text) and the topic vector (generated using the proposed approach) to the Word2vec/Doc2vec model. Once the embedding vector is generated from the model, we can use any similarity measure (explained already), between the sentences embedding vector and topic vector's embedding vector.

4.4. Sentence ranking and summary generation

On the basis of relevance generated in the previous step, top-ranked K sentences are included in the summary, where K is the number of sentences to be included in the summary and it is user-defined.

More than one identical sentence will result in the same quantified value of similarity with the topic vector. To remove the duplicate sentences, we skip those similar ranked sentences.

With the help of an example, now we describe LDA topic vector generation process. Suppose we have a news article given as Fig. 3 and we have to generate the topic vector for it. Let us assume; two topics have to be generated from the given text. Once we apply LDA methodology on the input text, we find the topic and respective words as given in Fig. 3. So it is very well understood that the first topic contains the words related to telecoms service charges, and the second topic contains the words related to cabinet policy. Once applied, the proposed method will include the relevant sentences in summary, that cover most of these topics words.

- **Different ways to create the topic vector:**

One of the important steps of the proposed methodology is sentence relevance finding, which can be achieved by creating the topic vector through the following two scenarios. Fig. 4 contains the details about both the scenarios (or variants).

Combined topic vector approach: Generating different topics and concatenating the words associated with those topics in a single topic vector, i.e. resulting T_V. Sentences of the input document are then compared with this combined topic vector T_V. Top most similar K sentences are included in the summary.

Individual topic vector approach: Generating different topic vectors (i.e., respective words of separate topics) resulting in T_{V1}, $T_{V2} \ldots T_{Vk}$. The sentences which are most similar to K different topic vectors respectively are included in the summary.

In the Individual topic vector variant of the proposed technique, the number of the topics should be equal to the number of sentences in summary (i.e., P should be equal to K). The sentences which are most similar to the topic vectors are kept in summary, and K is the threshold on how many sentences to be included in the summary. On the other hand, in the Combined topic vector variant of the proposed technique, the number of topics, i.e., P, can be independent of K. In our evaluation, we have kept the number of topics equal to the number of sentences to be included in the summary.

As an example, now we apply the topic vector generation step using the text given in Fig. 3. Using the Combined topic vector variant, the topic vector T_V would be generated, which is the set of all words present in both the topics (i.e., topic 1 and topic 2 in Fig. 3). If we use Individual topic vector variant, then there would be two separate topic vector T_{V1} (set of words in topic 1) and T_{V2} (set of words in topic 2).

Algorithm 1 includes the general steps of the proposed method. Although the given algorithm follows the general steps of the proposed methodology, the topic vector generation steps of the algorithm can be modified according to Fig. 4, for the two variants of the proposed method.

Algorithm 1 Topic-Based Text Summarization Method

Input

D: Text document with n sentences. Where $D = \{S_1, S_2 \ldots S_n\}$

P: The number of topics to be selected. *(User-defined)*

W: The number of words to be selected from each topic. *(User-defined)*

K: The number of sentences to be included in the summary. *(User Defined)*

Output

S: The Extractive summary of the given document. Where $S \subset D$

Steps

1: Decompose the given document into the set of sentences and perform

  1.1: Stop word removal

  1.2: Punctuation removal

  1.3: Lemmatization

2: Generate the vector for the sentences of the input text.

3: Use Topic Modeling to generate the topic vector (top W words selected from P topics, i.e., total $P * W$ words) of the document.

4: Find the similarity of each of the sentence vector with the topic vector.

5: For 1 to K repeat the following

  5.1: Include top-ranked sentence in the summary.

  5.2: Remove the included sentence from the input text.

  5.3: If a sentence have same rank as previous sentence then skip it.

5. Evaluation

The evaluation of the proposed methodology is performed on the CNN/ DailyMail (Hermann et al., 2015) and Opinosis (Ganesan et al., 2010) datasets for single document summarization. The results show the out-performance of the proposed methodology when compared with existing text summarization techniques.

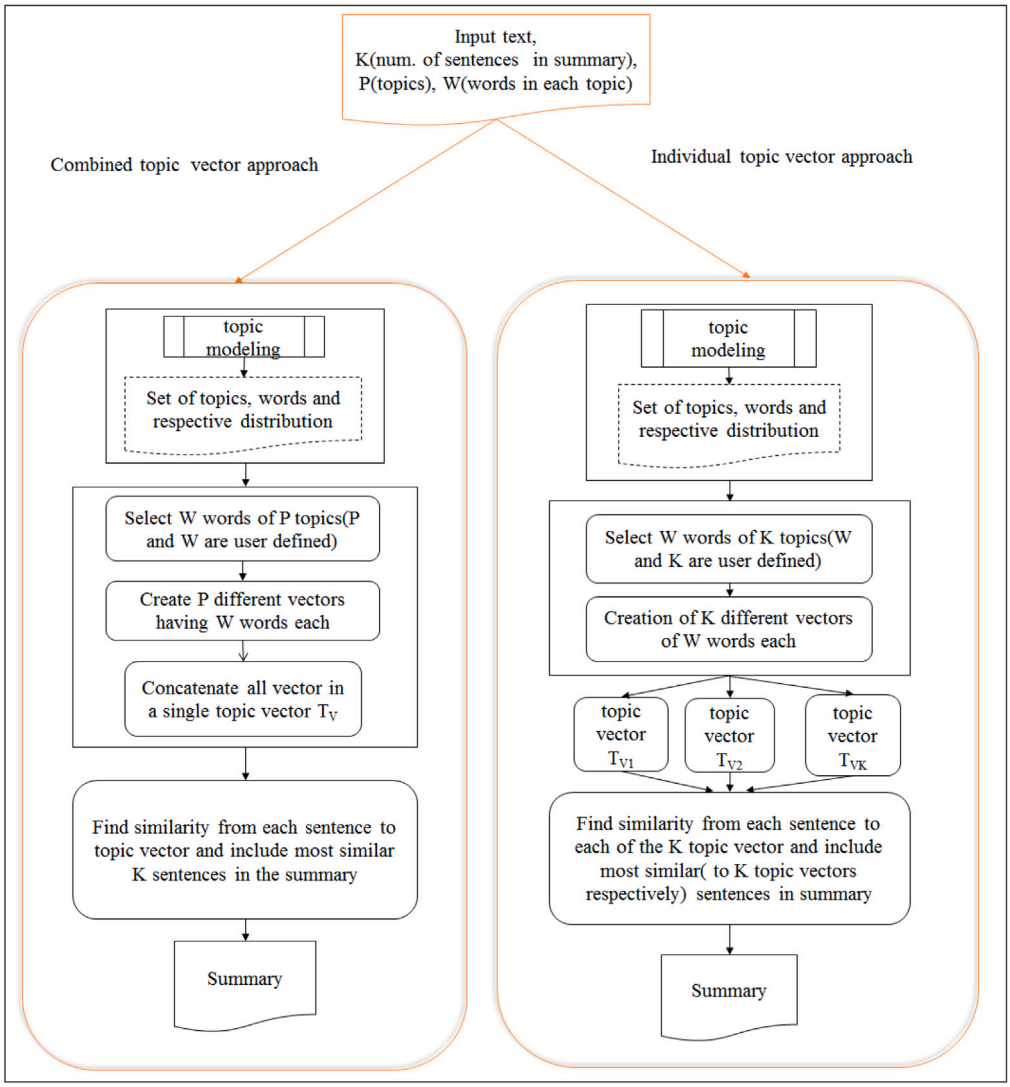

Fig. 4. Different ways to create topic vectors of the proposed method.

5.1. Evaluation parameters

We had compared the proposed method with the existing state-of-the-art text summarization techniques on ROUGE (Lin, 2004) parameters. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used for evaluating automatic summarization of the text data. Mostly used ROUGE matrices for evaluation are

ROUGE-1: Gives the overlap of 1-gram (single word) between the system and reference summaries.
ROUGE-2: Gives the overlap of bi-grams between the system and reference summaries.
ROUGE-N: Gives the overlap of N-grams between the system and reference summaries.
ROUGE-L: Used for finding the Longest Common Subsequence between the system and reference summaries.
R-SU4: Used to represent skip-bigrams based statistics.

In addition, P, R, and F with ROUGE refer precision, recall, and F-measure respectively. There parameters can be defined as below (Mirshojaee, Masoumi, & Zeinali, 2020)

$$Precision = \frac{Relvant_{Sentences} \cap Retrieved_{Sentences}}{Retrieved_{Sentences}}$$

$$Recall = \frac{Relvant_{Sentences} \cap Retrieved_{Sentences}}{Relevant_{Sentences}}$$

$$F - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

As we know, for automatic evaluation, we need a reference summary (or gold summary) to evaluate the proposed method on various ROUGE parameters. But there are some other works like Louis and Nenkova (2013) have introduced efficient summary

Table 1
Comparison among the different text summarization methods for F-Measure (CNN/DailyMail dataset).

Methods	R-1	R-2	R-L
RM (Gong & Liu, 2001)	0.355	0.130	0.173
TextRank (Mihalcea, 2004)	0.383	0.145	0.196
LexRank (Erkan & Radev, 2004)	0.385	0.140	0.207
LSA (Ozsoy et al., 2011)	0.344	0.122	0.170
TextRank Variation (Barrios et al., 2015)	0.391	0.150	0.192
Sumrunner (Nallapati et al., 2017)	0.396	0.162	0.352
REFRESH (Narayan et al., 2018)	0.400	0.182	0.366
LATENT (Zhang, Wu, Bu, Jiang, & Cao, 2018)	0.410	0.187	0.375
SUMO (Liu et al., 2019)	0.410	0.184	0.372
VHTM (Fu et al., 2020)	0.405	0.180	0.371
Proposed method (Individual topic vector)	0.401	0.200	0.391
Proposed method (Combined topic vector)	0.423	0.203	0.395

Table 2
Comparison among the different text summarization methods for maximum value of F-Measure (Opinosis dataset).

Methods	R-1	R-2	R-L	R-SU4
RM (Gong & Liu, 2001)	0.153	0.031	0.059	0.090
TextRank (Mihalcea, 2004)	0.193	0.044	0.075	0.112
LexRank (Erkan & Radev, 2004)	0.269	0.083	0.152	0.130
LSA (Ozsoy et al., 2011)	0.167	0.033	0.066	0.085
TextRank Variation (Barrios et al., 2015)	0.177	0.040	0.073	0.078
Proposed method (Individual topic vector)	0.189	0.080	0.153	0.110
Proposed method (Combined topic vector)	0.270	0.089	0.163	0.142

assessment measures, which require very little or no human input. On the other-hand Saggion, Torres-Moreno, da Cunha, SanJuan, and Velázquez-Morales (2010) have reviewed the association of text system's rankings using methods of evaluation with and without human models.

In the proposed method, we have used the ROUGE evaluation system (Lin, 2004) as to be applied on two standard datasets i.e., CNN/DailyMail (Hermann et al., 2015), and Opinosis (Ganesan et al., 2010) that are publicly available.

5.2. Corpus used

Two standard datasets are used for the evaluation of the proposed methodology. The first dataset is Opinosis dataset (Ganesan et al., 2010) comprising of 51 documents containing the sentences extracted from user reviews on the given topic collected from Tripadvisor (hotels), Edmunds.com (cars) and Amazon.com (various electronics). There are 51 topics (on average 100 sentences per topic). For each of the documents, five gold summaries are given. The second dataset is CNN/DailyMail dataset (Hermann et al., 2015) comprising of stories related to social, political and technical articles collected from CNN/DailyMail news. Kyunghyun Cho, academician (New York University) has made the dataset available for download, which can be easily accessed. In CNN/DailyMail dataset, each article contains the highlights which serve the purpose of summary for that article.

The reason for using two datasets is to prove the generalized property of the proposed method. The summary of the first dataset (Opinosis) is too abstract or short and of the second dataset (CNN/DailyMail) is comparatively lengthy in size.

5.3. Result

We have compared the proposed method with several text summarization methods to evaluate the effectiveness. Those methods are RM (Relevance Measure) method (Gong & Liu, 2001), TextRank method (Mihalcea, 2004), LexRank method (Erkan & Radev, 2004), LSA method (Ozsoy et al., 2011), TextRank Variation Method (Barrios et al., 2015), Sumrunner (Nallapati et al., 2017), REFRESH (Narayan et al., 2018), LATENT (Zhang, Lapata, Wei, & Zhou, 2018), SUMO (Liu, Titov, & Lapata, 2019) and VHTM (Fu et al., 2020).

We have evaluated the results for both variants of the proposed method (i.e., Individual topic vector and Combined topic vector approach). Table 1 shows the comparison among the different methods of the text summarization evaluated for ROUGE parameters on CNN/DailyMail dataset.

Table 2 shows the comparison among the different methods of the text summarization evaluated for maximum (out of five gold summaries for a particular text) value of ROUGE parameters on Opinosis dataset.

Table 3 shows the comparison among the different methods of the text summarization evaluated for average (of five gold summaries for a particular text) value of ROUGE parameters on Opinosis dataset. Here average stands for the average ROUGE score out of the five gold summaries given for each input text.

All the eleven methods have been evaluated on CNN/DailyMail (Hermann et al., 2015) dataset. We have compared six methods on the Opinosis (Ganesan et al., 2010) dataset because the remaining five methods are using learning techniques that need a massive

Table 3
Comparison among the different text summarization methods for average value of F-Measure, Precision and Recall (Opinosis dataset).

Methods	F-Measure	Precision	Recall
RM (Gong & Liu, 2001)	0.104	0.062	0.413
TextRank (Mihalcea, 2004)	0.133	0.085	0.382
LexRank (Erkan & Radev, 2004)	0.190	0.148	0.331
LSA (Ozsoy et al., 2011)	0.112	0.069	0.387
TextRank Variation (Barrios et al., 2015)	0.122	0.075	0.392
Proposed method (Individual topic vector)	0.145	0.089	0.393
Proposed method (Combined topic vector)	0.192	0.125	0.420

Table 4
Human evaluation results (Opinosis dataset).

Methods	Informativeness	Coherence
RM (Gong & Liu, 2001)	3.4	4
TextRank (Mihalcea, 2004)	3.3	3.5
LexRank (Erkan & Radev, 2004)	3.5	3.5
LSA (Ozsoy et al., 2011)	3.0	4.0
TextRank Variation (Barrios et al., 2015)	3.5	3.6
Proposed method (Individual topic vector)	3.8	3.7
Proposed method (Combined topic vector)	4.3	4.1

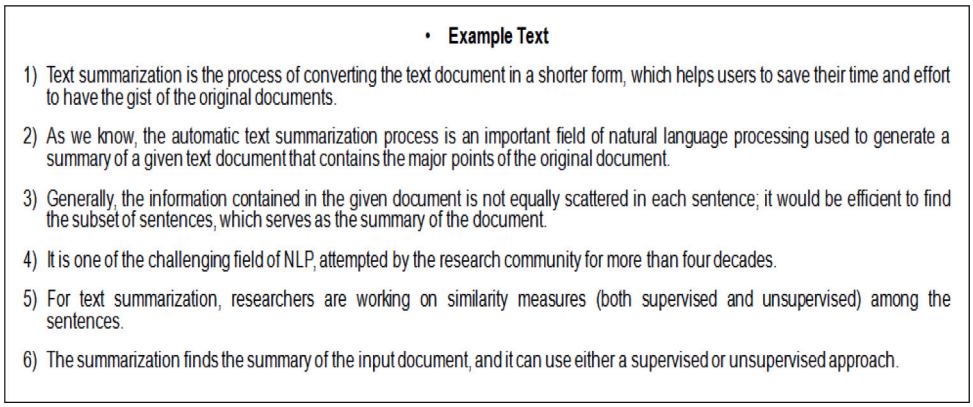

Fig. 5. Example input.

amount of the data to learn the mapping between input text and respective summary. Only CNN/DailyMail dataset contains a sufficient number of the instances to be used in learning techniques, whereas Opinosis dataset comprising of 51 documents and five gold summaries for each document. Moreover, Opinosis dataset is adequate for the remaining techniques.

As it is well known, the ROUGE matrices have wide acceptance in assessing text summarization systems. In the proposed work, we also perform the human evaluation to estimate the informativeness and coherence of the summaries that are generated by the proposed method and the compared methods. Like, Yang et al. (2020), researchers have asked to give each of the generated summary an integer score of 1 to 5 for informativeness, and coherence, separately. Three researchers, of which two are faculty members are asked to give each produced summary an integer score of 1 (bad), 2 (below average), 3 (average), 4 (satisfactory), 5 (very good) for informativeness and coherence, separately. For the coherence aspect, we check logical relations between the sentences, of the given text. In contrast, for the informativeness part, we check if the summary produced contains the salient information of the article. The human evaluation results are listed in Table 4. The proposed method outperforms the other approaches by a significant margin on Opinosis (Ganesan et al., 2010) dataset.

In the Combined topic vector approach, the topic vector is generated in such a manner that it includes the best words belonging to the various topic within the whole document. Then each sentence is assigned a rank according to semantic similarity with the topic vector. Consequently, according to calculated ranks, the logically associated sentences are automatically included in the summary. Whereas on the other hand, in the Individual topic vector approach, every sentence selected in summary automatically presents the input text's single topic. In this way, the order between the sentences is of little importance.

As an example, suppose we have text document containing the sentences given in Fig. 5

Now, if we want to find the sentence, which is the best candidate to be included in the summary. In a normal scenario, which is based on similarity with the overall document, "**Sentence 2**" is assigned as the highest rank because it is lengthy and covers most of the words of the overall document.

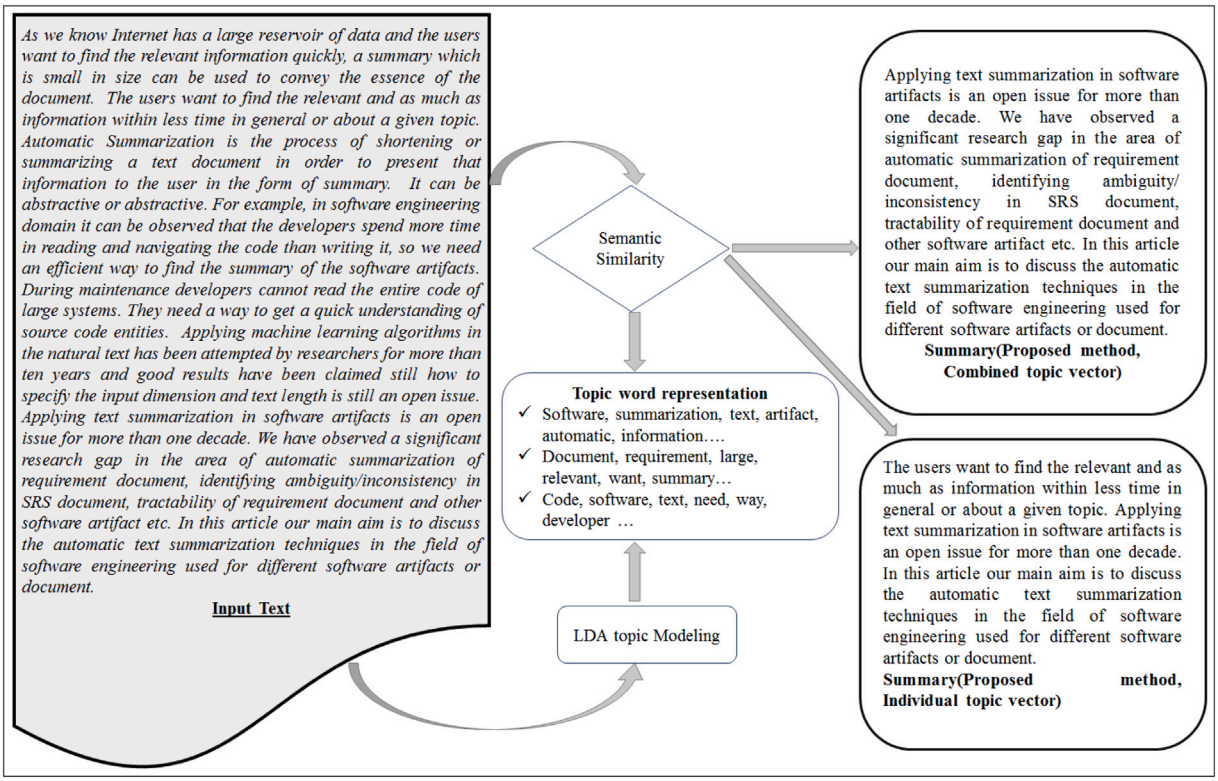

Fig. 6. Example summary generated by the proposed method.

Proposed methodology assigns the highest rank to "**Sentence 6**" because LDA methodology has generated, "summary", "supervised", "unsupervised", "process" and "summarization" as topic words with high probabilities and, "**Sentence 6**" has the highest similarity with the topic vector generated with those topic words. So it can be observed from the above-said example that our methodology includes in the summary those sentences which cover most of the topic words of the input document even if the sentence is comparatively smaller in size.

Fig. 6 shows the summaries generated by both variants of the proposed method.

So it is well observed from the evaluation that the summaries generated by the proposed methodology score better performance in terms of ROUGE parameters.

If we consider the complexity, our approach performs significantly better than existing text summarization techniques. In graph-based methods, every sentence's similarity has to be calculated with the rest of the sentences. The rank is then calculated with the help of graph-ranking methods, which is itself a time-consuming process. Whereas in the proposed technique, once the topic words have been generated, every sentence is compared with the topic vector only once. On the other hand, learning-based approaches (i.e., RNN, CNN, LSTM, etc.) have the main drawback: it takes a lot of time to learn the weight to train the model.

For evaluation, we can also use any language other than English. As far as creation of topic words for a language is concerned, LDA analyzes the ratio of each word's occurrence in the given document to the occurrence of the same word in all the documents. The procedure behind the topic creation approach is strictly mathematical and based on probability distribution.

Through evaluation, we have also proved that if a sentence is comparatively very small, but contains the main topic words of the overall document is included in the summary.

With the help of the proposed method's combined vector approach, we have shown that the sentence covering most of the topics representing the input document is assigned a higher rank. As a result, even more, topics can be covered in fewer sentences, which satisfy the summary's completeness property.

6. Conclusion and future work

This paper presents a topic-based extractive text summarization technique targeting those sentences to be included in the summary, which cover the maximum of the topics embedded in the input document. We address the redundancy issue by incorporating the topic modeling and semantic methods of similarity in the VSM method. The proposed method gives an excellent performance because the relevance of the sentence is not dependent on the similarity with the overall document; rather, sentences

are compared for semantic similarity with the topic vector (comprises of topic words of the original text). We have introduced two different methods to achieve our goal.

Most of the existing methods, directly or indirectly, use the document's key topics to get the best summary sentences. We find the principal problem with these methods is redundancy, i.e., the summaries usually contain several sentences conveying knowledge of the same sort. We structured the Individual topic vector approach of the proposed method so that each selected sentence for the summary should automatically represent only one topic of the given text. Whereas in the Combined topic vector approach of the proposed method, every sentence is chosen to address most of the topics. As a result, in summary, the likelihood of including redundant sentences is significantly reduced. Incorporating the semantic meanings of words in the similarity measure has also led to a significant improvement in the summary quality.

The result of ROUGE parameters obtained shows that our method outperforms over other text summarization methods.

In addition to automated evaluation, we have shown by the manual assessment that the proposed approach has considerable improvement over current state-of-the-art summarization methods. For manual evaluation, the summary has been assessed on informativeness and coherence parameters.

The main contribution of the proposed method is to present a generalized mechanism that substitutes the input document in the reduced-sized dimension (i.e., topic vector) to discover the relevance of the sentences. Rather than comparing with the overall document, sentences of the input text can be compared with the topic vector, resulting in the attractive results in terms of rouge parameters. Moreover, we have also incorporated the semantic measure to find the relevance of the sentences to be included in the extractive summary of the given document. We have kept topic generation and relevance finding steps independent, so that method remains flexible and adaptable for future changes.

We have implemented and evaluated the proposed method for the natural English language. Not that our approaches can only be applied for the English language, but it can be adapted for any other language as well. We have designed the topic generation and rank generation step in the proposed framework in a manner such that it can work for any language other than English.

In the future, we would like to incorporate topic modeling methodology in other techniques of extractive text summarization like graph-based methods.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdi, A., Shamsuddin, S. M., & Aliguliyev, R. M. (2018). QMOS: Query-based multi-documents opinion-oriented summarization. *Information Processing & Management*, *54*(2), 318–338.

Abdi, A., Shamsuddin, S. M., Hasan, S., & Piran, J. (2018). Machine learning-based multi-documents sentiment-oriented summarization using linguistic treatment. *Expert Systems with Applications*, *109*, 66–85. http://dx.doi.org/10.1016/j.eswa.2018.05.010.

Aggarwal, C. C. (2018). Information extraction. In *Machine learning for text* (pp. 361–380). Springer, http://dx.doi.org/10.1007/978-3-319-73531-3_12.

Alguliev, R. M., Aliguliyev, R. M., Hajirahimova, M. S., & Mehdiyev, C. A. (2011). MCMR: Maximum coverage and minimum redundant text summarization model. *Expert Systems with Applications*, *38*(12), 14514–14522. http://dx.doi.org/10.1016/j.eswa.2011.05.033.

Amplayo, R. K., & Song, M. (2017). An adaptable fine-grained sentiment analysis for summarization of multiple short online reviews. *Data & Knowledge Engineering*, *110*, 54–67.

Barrios, F., López, F., Argerich, L., & Wachenchauzer, R. (2015). Variations of the similarity function of textrank for automated summarization. In *Argentine symposium on artificial intelligence*.

Barros, C., Lloret, E., Saquete, E., & Navarro-Colorado, B. (2019). NATSUM: Narrative abstractive summarization through cross-document timeline generation. *Information Processing & Management*, *56*(5), 1775–1793.

Barzilay, R., & McKeown, K. R. (2005). Sentence fusion for multidocument news summarization. *Computational Linguistics*, *31*(3), 297–328. http://dx.doi.org/10.1162/089120105774321091.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022.

Chang, Y. -L., & Chien, J. -T. (2009). Latent Dirichlet learning for document summarization. In *Acoustics, speech and signal processing, 2009. IEEE international conference on* (pp. 1689–1692). IEEE, http://dx.doi.org/10.1109/icassp.2009.4959927.

Chen, W., Cai, F., Chen, H., & de Rijke, M. (2019). Hierarchical neural query suggestion with an attention mechanism. *Information Processing & Management*, Article 102040.

Cuong, H. -N., Tran, V. -D., Van, L. N., & Than, K. (2019). Eliminating overfitting of probabilistic topic models on short and noisy text: The role of dropout. *International Journal of Approximate Reasoning*.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479.

Fattah, M. A., & Ren, F. (2008). Automatic text summarization. *World Academy of Science, Engineering and Technology*, *37*, 2008.

Fattah, M. A., & Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech and Language*, *23*(1), 126–144. http://dx.doi.org/10.1016/j.csl.2008.04.002.

Ferreira, R., de Souza Cabral, L., Lins, R. D., e Silva, G. P., Freitas, F., Cavalcanti, G. D., et al. (2013). Assessing sentence scoring techniques for extractive text summarization. *Expert Systems with Applications*, *40*(14), 5755–5764. http://dx.doi.org/10.1016/j.eswa.2013.04.023.

Fu, X., Wang, J., Zhang, J., Wei, J., & Yang, Z. (2020). Document summarization with VHTM: Variational hierarchical topic-aware mechanism. In *AAAI* (pp. 7740–7747).

Fuad, T. A., Nayeem, M. T., Mahmud, A., & Chali, Y. (2019). Neural sentence fusion for diversity driven abstractive multi-document summarization. *Computer Speech and Language*, *58*, 216–230.

Ganesan, K., Zhai, C., & Han, J. (2010). Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics* (pp. 340–348). Association for Computational Linguistics.

Gong, Y., & Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 19–25). ACM, http://dx.doi.org/10.1145/383952.383955.

Gupta, V., & Lehal, G. S. (2010). A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, *2*(3), 258–268. http://dx.doi.org/10.4304/jetwi.2.3.258-268.

Gupta, P., Pendluri, V. S., & Vats, I. (2011). Summarizing text by ranking text units according to shallow linguistic features. In *Advanced communication technology, 2011 13th international conference on* (pp. 1620–1625). IEEE.

Haiduc, S., Aponte, J., Moreno, L., & Marcus, A. (2010). On the use of automated text summarization techniques for summarizing source code. In *Reverse engineering, 2010 17th working conference on* (pp. 35–44). IEEE, http://dx.doi.org/10.1109/wcre.2010.13.

Harabagiu, S. M., Lacatusu, V., & Morarescu, P. (2002). Multidocument summarization with GISTexter. In *LREC: Vol. 1*, (pp. 1456–1463). Citeseer.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., et al. (2015). Teaching machines to read and comprehend. In *Advances in neural information processing systems* (pp. 1693–1701).

Hu, M., Sun, A., & Lim, E. -P. (2008). Comments-oriented document summarization: Understanding documents with readers' feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 291–298). ACM, http://dx.doi.org/10.1145/1390334.1390385.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, *46*(5), 604–632. http://dx.doi.org/10.1515/9781400841356.514.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196).

Li, Y., McLean, D., Bandar, Z. A., Crockett, K., et al. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge & Data Engineering*, (8), 1138–1150. http://dx.doi.org/10.1109/tkde.2006.130.

Li, X., Wang, Y., Zhang, A., Li, C., Chi, J., & Ouyang, J. (2018). Filtering out the noise in short text topic modeling. *Information Sciences*, *456*, 83–96.

Lim, K. W., Buntine, W., Chen, C., & Du, L. (2016). Nonparametric Bayesian topic modelling with the hierarchical Pitman–Yor processes. *International Journal of Approximate Reasoning*, *78*, 172–191.

Lin, C. -Y. (2004). Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out, http://aclweb.org/anthology/W04-1013.

Liu, Z., & Jansen, B. J. (2017). Identifying and predicting the desire to help in social question and answering. *Information Processing & Management*, *53*(2), 490–504.

Liu, Y., Titov, I., & Lapata, M. (2019). Single document summarization as tree induction. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, Volume 1 (Long and Short Papers)* (pp. 1745–1755).

Lloret, E., & Palomar, M. (2009). A gradual combination of features for building automatic summarisation systems. In *International conference on text, speech and dialogue* (pp. 16–23). Springer, http://dx.doi.org/10.1007/978-3-642-04208-9_6.

Louis, A., & Nenkova, A. (2013). Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, *39*(2), 267–300.

Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K. -F., et al. (2016). *Detecting rumors from microblogs with recurrent neural networks*. AAAI Press.

Mani, I., & Bloedorn, E. (1998). Machine learning of generic and user-focused summarization. In *AAAI/IAAI* (pp. 821–826).

Mao, X., Yang, H., Huang, S., Liu, Y., & Li, R. (2019). Extractive summarization using supervised and unsupervised learning. *Expert Systems with Applications*, *133*, 173–181.

Maybury, M. (1999). *Advances in automatic text summarization*. MIT press.

Mihalcea, R. (2004). Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL 2004 on interactive poster and demonstration sessions* (p. 20). Association for Computational Linguistics, http://dx.doi.org/10.3115/1219044.1219064.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Mirshojaee, S. H., Masoumi, B., & Zeinali, E. (2020). MAMHOA: A multi-agent meta-heuristic optimization algorithm with an approach for document summarization issues. *Journal of Ambient Intelligence and Humanized Computing*, 1–16.

Mutlu, B., Sezer, E. A., & Akcayol, M. (2019). Multi-document extractive text summarization: A comparative assessment on features. *Knowledge-Based Systems*, *183*, Article 104848.

Na, L., Ming-xia, L., Ying, L., Xiao-jun, T., Hai-wen, W., & Peng, X. (2014). Mixture of topic model for multi-document summarization. In *The 26th Chinese control and decision conference* (pp. 5168–5172). IEEE.

Nallapati, R., Zhai, F., & Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI conference on artificial intelligence*.

Narayan, S., Cohen, S. B., & Lapata, M. (2018). Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies, Volume 1 (Long Papers)* (pp, 1747–1759).

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In *Mining text data* (pp. 43–76). Springer, http://dx.doi.org/10.1007/978-1-4614-3223-4_3.

Nguyen, M. -T., Tran, V. C., Nguyen, X. H., & Nguyen, L. -M. (2019). Web document summarization by exploiting social context with matrix co-factorization. *Information Processing & Management*, *56*(3), 495–515.

Oyedotun, O. K., & Khashman, A. (2016). Document segmentation using textural features summarization and feedforward neural network. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *45*(1), 198–212.

Ozsoy, M. G., Alpaslan, F. N., & Cicekli, I. (2011). Text summarization using latent semantic analysis. *Journal of Information Science*, *37*(4), 405–417. http://dx.doi.org/10.1177/0165551511408848.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The pagerank citation ranking: bringing order to the web: Tech. rep.*, Stanford InfoLab.

Radev, D. R., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics*, *28*(4), 399–408. http://dx.doi.org/10.1162/089120102762671927.

Saggion, H., Torres-Moreno, J. -M., da Cunha, I., SanJuan, E., & Velázquez-Morales, P. (2010). Multilingual summarization evaluation without human models. In *Coling 2010: Posters* (pp. 1059–1067).

Salminen, J., Hopf, M., Chowdhury, S. A., Jung, S.-g., Almerekhi, H., & Jansen, B. J. (2020). Developing an online hate classifier for multiple social media platforms. *Human-centric Computing and Information Sciences*, *10*(1), 1.

Salton, G., Wong, A., & Yang, C. -S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620. http://dx.doi.org/10.1145/361219.361220.

Song, S., Huang, H., & Ruan, T. (2019). Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications*, *78*(1), 857–875.

Torres-Moreno, J. -M. (2014). *Automatic text summarization*. John Wiley & Sons.

Van Lierde, H., & Chow, T. W. (2019). Query-oriented text summarization based on hypergraph transversals. *Information Processing & Management*, *56*(4), 1317–1338.

Yang, M., Wang, X., Lu, Y., Lv, J., Shen, Y., & Li, C. (2020). Plausibility-promoting generative adversarial network for abstractive text summarization with multi-task constraint. *Information Sciences*, *521*, 46–61.

Yeh, J. -Y., Ke, H. -R., Yang, W. -P., & Meng, I. -H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing and Management*, *41*(1), 75–95.

Yousefi-Azar, M., & Hamey, L. (2017). Text summarization using unsupervised deep learning. *Expert Systems with Applications*, *68*, 93–105.

Zhang, X., Lapata, M., Wei, F., & Zhou, M. (2018). Neural latent extractive document summarization. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 779–784).

Zhang, L., Wu, Z., Bu, Z., Jiang, Y., & Cao, J. (2018). A pattern-based topic detection and analysis system on Chinese tweets. *Journal of Computational Science*, *28*, 369–381.