



# A new graph-based extractive text summarization using keywords or topic modeling

Ramesh Chandra Belwal<sup>1</sup> · Sawan Rai<sup>1</sup> · Atul Gupta<sup>1</sup>

Received: 20 April 2020 / Accepted: 3 October 2020 / Published online: 17 October 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

In graph-based extractive text summarization techniques, the weight assigned to the edges of the graph is the crucial parameter for the sentence ranking. The weights associated with the edges are based on the similarity between sentences (nodes). Most of the graph-based techniques use the common words based similarity measure to assign the weight. In this paper, we propose a new graph-based summarization technique, which, besides taking into account the similarity among the individual sentences, also considers the similarity between the sentences and the overall (input) document. While assigning the weight among the edges of the graph, we consider two attributes. The first attribute is the similarity among the nodes, which forms the edges of the graph. The second attribute is the weight given to a component that represents how much the particular edge is similar to the topics of the overall document for which we incorporate the topic modeling. Along with these modifications, we use the semantic measure to find the similarity among the nodes. The evaluation results of the proposed method demonstrate a significant improvement of the summary quality over the existing text summarization techniques.

**Keywords** Text summarization · Extractive summarization · Graph-based · Topic-based · Similarity measure

## 1 Introduction

The amount of the data available on the Internet has achieved such an enormous volume; it is infeasible for human beings to extract useful information within the desired time. Without summaries, it is impractical for the users to read the vast information which is available online (Saggion and Poibeau 2013). Hence we need a method through which we can have the essence of the large text effectively at the desired time.

The text summarization is the method of creating the compressed or shorter version of a given text document that contains useful information for the users. The fundamental aim of the text summarization is to reduce the content

and size of the given text to its important points (Alterman 1991).

Using computer algorithms, the summarization methods produce a summary of given text while retaining the original meaning (Mirshojaee et al. 2020).

The text summarization can be categorized based on various parameters. On the basis of the output type, the summarization can be either abstractive or extractive. The extractive summarizers produce the summaries by selecting a few relevant or important sentences from the original document. In the abstractive summarization, the summary is generated, which consists of ideas or concepts taken from the original document. The abstractive summarizers may even produce new sentences to the summary.

To find the best sentence for the summary, the summarization techniques can use either a query-based or generic approach. In query-based methods the sentences for summary are generated based on given topic or query (Ouyang et al. 2011; Abdi et al. 2018a; Van Lierde and Chow 2019; Yousefi-Azar and Hamey 2017). The proposed method is generic approach of summarization where generalized summary is generated irrespective of any query.

Various summarization methods are proposed for both the extractive and abstractive summarizations. Text

✉ Ramesh Chandra Belwal  
rameshbelwal@gmail.com

Sawan Rai  
sawanrai@iiitdmj.ac.in

Atul Gupta  
atul@iiitdmj.ac.in

<sup>1</sup> Department of Computer Science and Engineering,  
Indian Institute of Information Technology Design  
and Manufacturing, Jabalpur, India

summarization originally started with feature -based strategies where different attributes (linguistic and statistical) are used to quantify the importance of the sentences to be included in the summary (Baxendale 1958; Fattah and Ren 2008; Ferreira et al. 2013; Gupta et al. 2011; Lloret and Palomar 2009; Luhn 1958; Xu et al. 2017; Mutlu et al. 2019; Abdi et al. 2018b). Relevance Measure based (RM-based) methods (Gong and Liu 2001; Haiduc et al. 2010) use statistical similarity to find the ranks of the sentences to be included in the summary. Topic-based methods (Blei et al. 2003; Chang and Chien 2009; Fu et al. 2020; Luhn 1958; Nguyen et al. 2019) use the distribution of the topic words embedded in the input document to find the important sentences for the summary. Latent Semantic Analysis based methods (LSA-based) (Gong and Liu 2001; Haiduc et al. 2010; Ozsoy et al. 2011) use the singular value decomposition to reduce the dimension of the sentence vector to be included in the summary. Graph-based methods (Barrios et al. 2016; Erkan and Radev 2004; Ganesan et al. 2010; Mihalcea 2004; Mihalcea and Tarau 2004; Nguyen-Hoang et al. 2012; Thakkar et al. 2010) first create a graph for the input text; afterward graph ranking techniques are applied to rank the candidate sentences for the summary. Template-based methods (Harabagiu et al. 2002) use the templates populated with information extracted from the input document. Recently, machine learning based methods (Abdi et al. 2018b; Fuad et al. 2019; Iyer et al. 2016; Mani and Bloedorn 1998; Mao et al. 2019; Narayan et al. 2018; Rush et al. 2015; Yang et al. 2020; Zhang et al. 2018b) are extensively used for text summarization, where trainable summarizers find the parameters of the candidate sentences to be included in the summary.

In the topic-based techniques, each sentence is assumed to represent one of the themes/topics embedded in the given document. As a result, many times, more than one sentences selected in summary represent a similar topic or subject. As statical feature-based methods do not consider the semantic significance of words, many times, good sentences are overlooked to be included in the summary. Whereas, most abstractive methods require linguistic treatment for the implementation. If we observe the supervised techniques, these methods have dataset constraints, i.e., the prerequisite of the training data (both the input text and respective summary).

The problem associated with the extractive methods of the summarization is the redundancy, i.e., more than one sentences included in the summary usually contain a similar sort of information. On the other hand, abstractive text summarizers are very complex in nature.

Considering the above-discussed shortcomings, we incorporate the topic/keyword modeling in the graph-based method so that sentences representing the given text's best

topics can be identified efficiently. Moreover, rather than using a statistical approach, we utilized semantic measure while finding the similarity between the sentence in the graph. Apart from all this, we have proposed a new formula that incorporates topic words (or keywords) generation and weight assignment steps independently to make the algorithm adaptable for future changes.

The proposed methods is an unsupervised graph-based approach for extractive summarization, where our prime concern is to overcome the problem of redundancy by including only those sentences in the summary which contain the maximum topics of the input text.

In the graph-based methods, the sentences of the input document are represented as the nodes of the graph and various similarity measures are used to assign the weight to the edges of the graph (Barrios et al. 2016; Erkan and Radev 2004; Mihalcea 2004; Mihalcea and Tarau 2004; Thakkar et al. 2010). The graph ranking algorithms (Brin and Page 1998; Herings et al. 2001; Kleinberg 1999) find the rank of nodes so that top-ranked nodes (i.e., sentences) can be included in the extractive summary of the input document. The graph-based extractive text summarization techniques use the following generalized steps

- Representation of the sentences in the form of a graph (sentences as nodes and similarity measure to assign a weight to the edges)
- Applying graph-based methods to find the ranks of the nodes.
- Summary generation on the basis of rank.

Assigning the weight to edges of the graph is the challenging task for the graph-based text summarization techniques. The similarity measure introduced in the proposed method is dependent on both the sentences ( which form the edges) as well as the overall (input) document.

Evaluation of the proposed method is carried out on two publicly available datasets, i.e., CNN/DailyMail (Hermann et al. 2015) and Opinosis (Ganesan et al. 2010). The proposed approach is evaluated on both the datasets for various ROUGE tools (Lin 2004) and compared with existing state-of-the-art methods of text summarization. The summary generated by the proposed method is found to score better results when compared to existing text summarization techniques.

The remaining of the paper is organized as follows: Sect. 2 includes a detailed discussion on related text summarization methods, where we have concentrated explicitly on graph-based techniques. Section 3 has a detailed discussion about the proposed method. The evaluation of the proposed method is demonstrated in Sect. 4 followed by the conclusion and future work in Sect. 5.

## 2 Related work

The proposed method is based on graph-based extractive text summarization techniques. The fundamental idea behind the extractive strategy of the text summarization is to discover the importance (or relevance) of the sentences so that the best sentences for the summary can be identified. The text summarization methods mainly differ according to how the sentences of the input document are represented in intermediate form, and the rank (importance) of the sentences is decided. Correlating the proposed work with the techniques available in the field of text summarization, we have organized the related works in the manner as given in Fig. 1. We have mainly focused on extractive text summarization

techniques. As the proposed work is based on graph-based techniques, we have an explicit and detailed analysis of graph-based text summarization.

The extractive summarization techniques produce the summaries by selecting a subset of the sentences from the original document (Allahyari et al. 2017). Text summarization techniques originally started by utilizing the different feature incorporated in the text data to find the score of the sentences to be included in the summary. In feature-based summarization, numerous statistical and linguistic features are used to find the relevance (or importance) of the sentences to be included in the summary. Various features have been used by researchers for assigning relevance to the sentences, i.e., word frequency, sentence position, proper noun, word co-occurrence, tf-idf, lexical similarity,

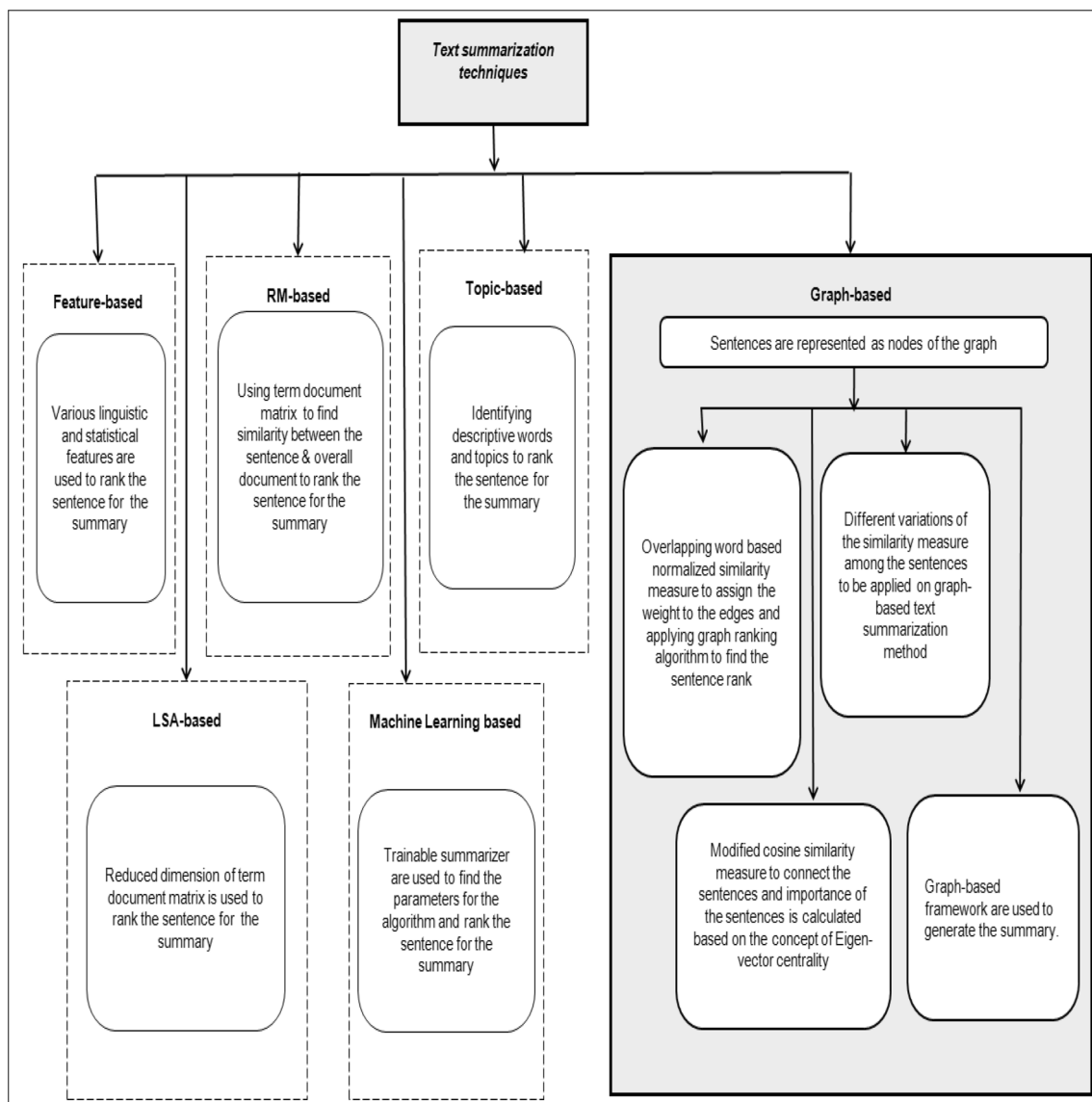


Fig. 1 Related work

sentence resemblance to title, cue phrases, etc (Baxendale 1958; Edmundson 1969; Fattah 2014; Fattah and Ren 2008; Ferreira et al. 2013; Gupta et al. 2011; Lloret and Palomar 2009; Mutlu et al. 2019; Xu et al. 2017). Which of the feature set will be the best combination for the summarization algorithms, is still a challenging task. The results of these methods are also affected because of the domain of the input text. The proposed method is independent of the domain of the text.

Relevance Measure (RM) based text summarization uses statistical distance among the sentence and overall document to find the best sentences for the summary. Gong and Liu (2001) proposed the RM and LSA based techniques to generate the extractive summary of the input document. The similarity between each sentence and the overall document is calculated with the help of statistical methods, and sentences are ranked on the basis of the similarity. The main step is the representation of the terms (i.e., words) and documents in the form of a matrix where each row represents a document (or sentence), and each column corresponds to a term or vice versa. The value of a cell in the matrix is the representation of the weight of a term (column) with respect to a document (the row). Gong and Liu (2001), in their work, have calculated the similarity of each sentence with the overall document vector, and rank has been generated based on similarity. The top-ranked  $K$  sentences are included in the extractive summary of the input document. The statistical measures are used to find the relevance of the sentences in RM based techniques. On the other hand, we have incorporated semantic measures for the similarity calculation among the sentences.

The dimension reduction techniques are applicable in text data that operate on the modified term-document matrix. Latent Semantic Analysis technique uses the concept of singular value decomposition to reduce the dimension (Deerwester et al. 1990). LSA is the technique used to reduce the number of rows while preserving the similarity among columns. Gong and Liu (2001) and Ozsoy et al. (2011) used Latent Semantic Analysis (LSA) based text summarization in their work.

The proposed method is based on a graph-based strategy, where we have incorporated the topic modeling technique to create the graph. We have an explicit discussion on graph-based techniques where our main objective is to investigate how the weights of the edges in the sentence-based graphs methods can affect the results of the text summarization.

Graph-based ranking algorithms are used to decide the importance of the nodes (vertices) within a graph, by utilizing the information drawn from the graph structure. The graph-based algorithms are also applicable in text-based ranking applications.

In graph-based techniques, the sentences are considered as nodes and common words based methodology is utilized

to create the edges among the nodes (Vetriselvi and Gopalan 2020).

Kleinberg (1999) introduced Hyperlink-Induced Topic Search (HITS) is a graph-based algorithm that rates Web pages. He developed an algorithm that uses the web link structure in order to discover and rank pages that are relevant to a particular topic. For each node, HITS algorithm generates two sets of scores i.e. “authority” and “hub” as follows

$$HITS_{Authority}(V_I) = \sum_{V_j \in In(V_I)} HITS_{Hub}(V_j) \quad (1)$$

$$HITS_{Hub}(V_I) = \sum_{V_j \in Out(V_I)} HITS_{Authority}(V_j) \quad (2)$$

HITS algorithm has also been successfully applied in automatic text summarization task.

Herings et al. (2001) introduced the positional power method, a ranking algorithm determining the score of a vertex. The score of the vertex is dependent on both the number of its successors and respective scores.

$$POSITIONAL_{Power}(V_I) = \frac{1}{|V|} \sum_{V_j \in Out(V_I)} (1 + POSITIONAL_{Power}(V_j)) \quad (3)$$

PageRank is an algorithm for deciding the rank of web pages (Brin and Page 1998). The algorithm assigns a higher rank to a web page that is more frequently cited by other web pages. Let  $G = (V, E)$  is the directed graph, where  $V$  is the set of nodes (or vertices) and  $E$  is the set of edges.  $In(V_I)$  is the set of vertices that point to it (in-degree), and  $Out(V_I)$  is the set of vertices that the vertex  $V_I$  points to (out-degree). The rank (or score) of a vertex is calculated as follows

$$Rank(V_I) = (1 - d) + d * \sum_{V_j \in In(V_I)} \frac{1}{|Out(V_j)|} Rank(V_j) \quad (4)$$

or in the simplified form, it is like

$$Rank(V) = (1 - d) + d(Rank(V_1)/C(V_1) + \dots + Rank(V_N)/C(V_N)) \quad (5)$$

$Rank(V)$  is the rank of page  $V$ , and  $C(V)$  is the number of links out of the page  $V$ .

where  $d$  is a damping factor which shows the probability of jumping from a given vertex to another random vertex in the graph. It can be set the value from 0 to 1 and have the default value 0.85.

On the basis of PageRank, Mihalcea and Tarau (2004) proposed the TextRank method, which is used to predict the rank of text sentences to generate the summary. The sentences of the text are represented as the nodes, and the similarity between the sentences is represented as the edge

among the nodes. A function is needed to compute the similarity between the sentences and used to build edges in between. For identifying the edges that connect such text units or nodes, different methods, i.e., overlapping words among the sentences, cosine similarity, longest common subsequence (LCS), etc. can also be used. The edges of the graph can be directed or undirected. The generalized steps used in TextRank algorithms are as follows

- Step 1. Identify text units (as sentences) and add them as nodes in the graph.
- Step 2. Now identify the relations that will be used to connect nodes (text units), and use these relations to draw edges between nodes in the graph.
- Step 3. Use the graph-based ranking algorithm (i.e., PageRank) and iterate until convergence.
- Step 4. Sort the nodes based on their final rank score.

The final rank value associated with each node is used for sentence selection (in summary) decisions. Mihalcea and Tarau (2004) introduce a new formula for graph-based ranking algorithm.

$$Rank^w(V_I) = (1 - d) + d * \sum_{V_J \in In(V_I)} W_{JI} \frac{Rank^w(V_J)}{\sum_{K \in Out(V_J)} W_{KJ}} \quad (6)$$

In addition to the connection between the nodes, they have also incorporated the concept of weight to the edges. The weight  $W_{IJ}$  is the weight of the edge connecting the node  $V_I$  and  $V_J$ .

Different methods are used to assign a weight to the edges of the graph. One of the ways is to use the number of overlapping words between two sentences which are denoted by the nodes  $V_I$  and  $V_J$ . Actually, the weight of the edges is the representation of similarity measure between the sentences (or the nodes of the graph). Mihalcea and Tarau (2004) used the following measure for the similarity between sentences  $S_I$  and  $S_J$

$$Similarity(S_I, S_J) = \frac{|W_K| W_{K \in S_I} \& W_{K \in S_J}}{\log(|S_I|) + \log(|S_J|)} \quad (7)$$

The problem with the approach is that the node weight and semantic relationship among the text units have not been taken into account.

As the node weight has been concerned the  $TF - IDF$  is the most used technique by the researchers

$$TF - IDF(t, d, D) = TF(t, d).IDF(t, D) \quad (8)$$

where  $TF$  (Term Frequency) number of times that term  $t$  (or word) occurs in document  $d$ .

$IDF$ (Inverse Document Frequency) =  $\log \frac{N}{|d \in D: t \in d|}$  is the measure of how much information the term or word

provides, that is, whether the term is common or rare across all documents (in the multi-document scenario).

Bellaachia and Al-Dhelaan (2012) proposed NE-Rank (Node and Edge Rank), a modification on TextRank using the weight of nodes that resulted as a node-edge weighting approach as a new formula

$$R(V_I) = (1 - d) * W(V_I) + d * W(V_I) \sum_{J: V_J \rightarrow V_I} \frac{W_{JI}}{\sum_{K: V_J} W_{JK}} R(V_J) \quad (9)$$

where  $W(V_I)$  is the weight of the current node and normalized  $TF - IDF$  is used to assign the weight. Bellaachia and Al-Dhelaan (2012) used  $TF - IDF$  to compute the weight of the node of the given graph as

$$W(V_I)_{TFIDF} = TF(V_I) * \log \frac{N}{DF(V_I)} \quad (10)$$

Erkan and Radev (2004) have introduced a new concept for measuring the centrality or importance of the sentences. To define the similarity between the sentences (i.e.,  $X$  and  $Y$ ) they introduced the modified formula for  $TF - IDF$  cosine similarity as

$$idf\_modified\_cosine(XY) = \frac{\sum_{W \in X, Y} TF_{W, X} TF_{W, Y} (IDF_W)^2}{\sqrt{\sum_{X_i \in X} (TF_{X_i, X} IDF_{X_i})^2} * \sqrt{\sum_{Y_i \in Y} (TF_{Y_i, Y} IDF_{Y_i})^2}} \quad (11)$$

For defining the similarity, they used a bag-of-word model where each sentence represented as an n-dimensional vector, where n is the number of all possible words in the target language and TF, IDF are the term frequency and inverse document frequency respectively.

Barrios et al. (2016) proposed the alternative of similarity measure to produce significant improvements over the original TextRank algorithm. The similarity measures they proposed are orthogonal to the TextRank model so these are easily integrated into the algorithm. They found some of these variations to produce improvements over the original algorithm as follows

*LCS* Longest Common Substring of the texts has been used as a similarity among the nodes.

*Cosine Distance* Cosine similarity can be a measure to find the similarity among the sentences or nodes of the graph. It can be defined as

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

They Barrios et al. (2016) used a classical  $TF - IDF$  model to represent the documents as vectors.

*BM25/Okapi-BM25* Is a ranking function which is widely used for Information Retrieval tasks based on the

probabilistic model. It is a variation of  $TF - IDF$  model. Okapi BM25 is a ranking function which is used by search engines to rank matching documents as per their relevance to a given query. It is defined as:

Suppose we have given two sentences A, B, BM25 can be defined as

$$BM25(A, B) = \sum_{i=1}^n IDF(B_i) * \frac{f(B_i, A) * (K_1 + 1)}{f(B_i, A) + K_1 * (1 - b + b * \frac{|A|}{avgD_L})} \quad (12)$$

where  $k$  and  $b$  are the parameters.  $avgD_L$  is the average length of the sentence. Barrios et al. (2016) used parameter  $k = 1.2$  and  $b = 0.75$ .

In the proposed method, we incorporate the topic-based technique in graph-based extractive text summarization. Topic modeling is the technique where the document is assumed to be the collection of words and their respective topic distribution. The rank of the sentences is generated on the basis of their similarity with the topic embedded in the input document. Blei et al. (2003) introduced Latent Dirichlet Allocation (LDA) topic modeling technique to find the topics and respective word distribution in the given document. In the topic representation techniques, an intermediate form of the input text is generated, which captures the topics of the input text (Nenkova and McKeown 2012). Based on these topics, sentences in the input text are scored for importance. Arora and Ravindran (2008) used Latent Dirichlet Allocation for capturing the events covered within the documents and generated the summary containing those sentences representing different events.

In the proposed work we use LDA method to generate the topic words of the given text. Top  $W_T$  words are selected from  $T$  topics, where  $W_T$  and  $T$  are user defined. LDA is a topic modeling technique used to learn the representation of the topics and respective topic distribution in a given text document (Blei et al. 2003). LDA generates the topic representation as follows

1. In the first step, the number of topics are selected.
2. LDA goes through each of the words in every document, and randomly assigns the word to one of the topics. In this manner we have documents representation in terms of topics. The random assignment gives us topic representations of the given documents and word distributions of the topics (although it is not good assignment).
3. LDA checks through all the documents, the percentage of times that each word has been assigned to a particular topic. Now LDA calculate

$p(T|D)$  = percentage of words in document D that are currently assigned to topic T. and

$p(W_T|T)$  = percentage of times the word  $W_T$  has been assigned to T in overall documents.

4. Now reassigned  $W_T$  to a new topic, based on value  $p(T|D) * p(W_T|T)$
5. The above steps of topic assignment are repeated for every word in each document, iteratively till convergence. LDA generates the output as the topics and respective words associated with the topics. It also generates the probability of words associated with the generated topics.

Ali et al. (2020) have introduced topic and sentiment aware microblog summarization for the twitter text. Na et al. (2014) have introduced LDA(Latent Dirichlet Allocation) based, namely Titled-LDA method for multi-document summarization.

Topic modeling is used in various text analytic tasks, such as, Lim et al. (2016) applied Bayesian topic modeling in social media for modeling the text data. Cuong et al. (2019) have analyzed the various techniques to incorporate dropout in topic models to remove the over-fitting for short text. Amplayo and Song (2017) have introduced fine-grained sentiment analysis to summarize the short texts with the help of topic modeling. Zhang et al. (2018a) have proposed a pattern-based topic detection technique that summarizes the tweet data. Barros et al. (2019) have proposed NATSUM to generate the abstractive summary from various news documents representing the same topic. Li et al. (2018) have introduced common semantics topic model (CSTM) by introducing common topic, to gather the noisy words. Fu et al. (2020) have proposed a variational hierarchical model (VHTM) that address summarization with topic inference via encoder-decoder and merging topics into multi-grained level using topic embedding and attention.

Recently, methods based on machine learning techniques are widely used for summarization task. Extensively used techniques are based on CNN, RNN, Encoder Decoder, LSTM, etc. SummaRuNNer (Nallapati et al. 2017), an RNN based model is introduced by Nallapati et al. to generate the summary of the given text. Encoder with attention-based extractor framework is introduced by Narayan et al. (2017) that uses the attention mechanism over the side information. To generate the extractive summary, Narayan et al. (2018) have introduced the reinforcement learning based technique. Zhang et al. (2018b) have introduced a latent variable extractive method that leverage human-generated summaries in the summarization task. Lovinger et al. (2019) introduced unsupervised learning and sentiment analysis based method

(named Gist), that selects the sentences characterizing a best set of reviews.

All learning-based techniques need a huge amount of instances to learn the parameters for the summarization algorithms. While the proposed method can be easily implemented in a small number of instances very well.

If deeply analyzed, then all the techniques, somehow directly or indirectly, use the topic of the document to get the summary's best sentence. We find that redundancy is the main problem with these techniques, i.e., the summaries usually contain multiple sentences that convey the same type of information or represent the same topic.

On the other hand, some methods use the statistical approach to generate the relevance/weight of the sentences while ignoring the semantic meaning of words. If a sentence is too short but holds more than one topic, then how to include that sentence has not been explicitly discussed in the literature.

Graph-based methods suffer from the redundancy problem, i.e., more than one similar sentences having high score can be selected for the summary. Moreover, the candidate sentences for the summary are primarily affected by the accuracy of the similarity measure. The semantic meaning of the words has not been incorporated in the graph-based methods.

To overcome the above-said shortcomings, we have incorporated the topic modeling in graph-based methods. Only those sentences get the high score, which represent the topics embedded in the input document. While assigning the weight to the edges of the graph, the graph-based methods mainly consider the similarity among the nodes (i.e., sentences of the input document) only. In addition to the similarity among the nodes, we have focused on the topic (embedded in the document) also.

Ultimately, the proposed work is a new graph-based extractive text summarization method, where we have shown that the similarity between each sentence as well as the overall document also plays an important role in deciding the rank of the sentence. The next section includes a detailed explanation of the proposed method with suitable examples.

### 3 The proposed method

In graph-based text summarization, the sentences are represented in the form of nodes, and the relationship between the nodes is used to draw the edges between the nodes. The essential step for these methods is to assign the weights to the edges of the graph, where weight depends on the similarity between the nodes of the graph. In the proposed work, we have mainly targeted the following parameters

- The first parameter is the similarity measure that connects the text units (i.e. nodes) and used to assign the weight to the edges of the graph.
- The second parameter for the weight of the edges that is the similarity between the edge forming nodes (sentences) and topics embedded in the input document.

Along with the above modifications, we have also incorporated the semantic measure while computing the similarity among the nodes (i.e., sentences). Figure 2, includes all the steps followed by the proposed methodology.

The first step of the proposed approach is the preprocessing of the input document. The input document is initially split into a set of sentences, followed by cleaning of the text. We have mainly eliminated those tokens, which have little meaning associated with the input document context. We have removed stops words and semicolons, after which we lemmatized the words of the sentences. Once the input document sentences are preprocessed, a graph is created that utilizes sentences as the nodes of the graph. Any of the two nodes are connected if they have common words between them.

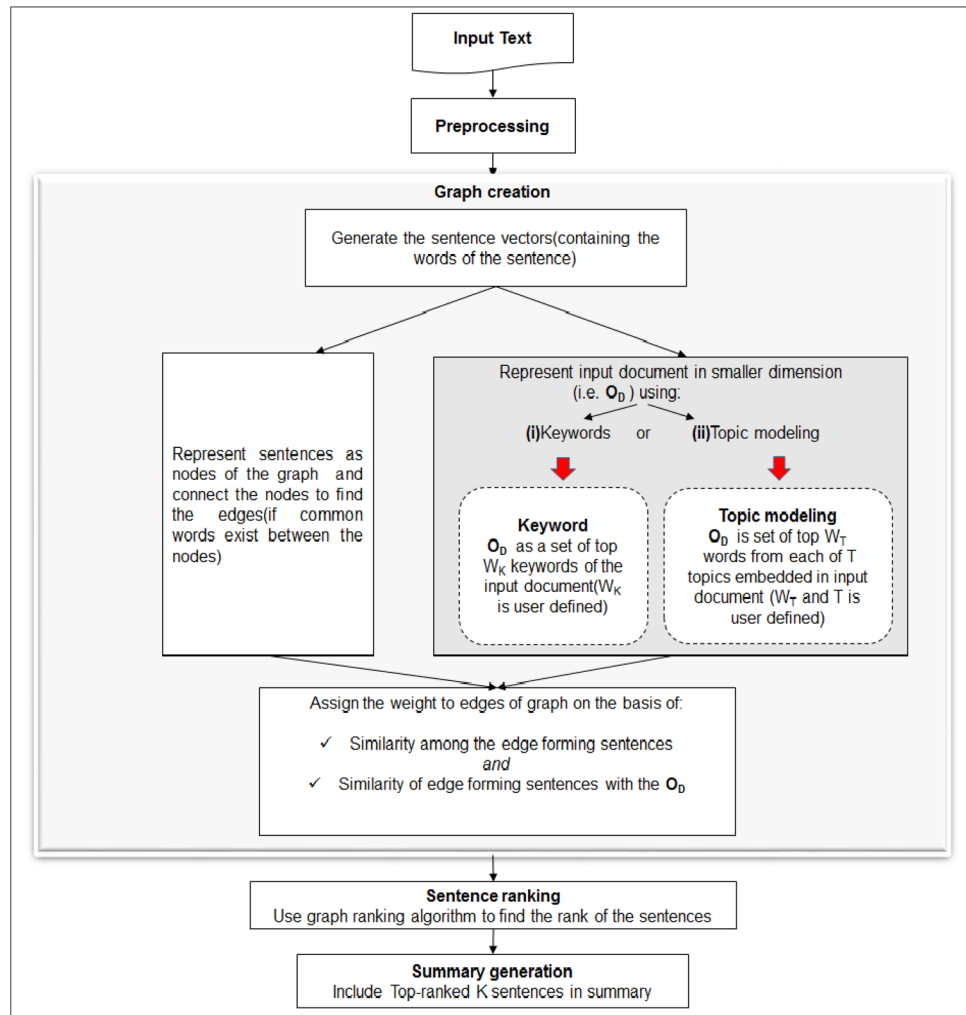
As the graph has been created, the remaining steps are the weight assignment to the edges and applying the graph ranking algorithm to find the ranks of the nodes. Once the ranks have been calculated, the extractive summary is generated based on ranks.

In graph-based text summarization, the weight assigned to the edge usually is dependent on the similarity between the connecting nodes (i.e., sentences). Various strategies have been proposed to assign the weights to the edges of the graph. These similarity measures (Barrios et al. 2016; Bellaachia and Al-Dhelaan 2012; Erkan and Radev 2004; Mihalcea 2004; Mihalcea and Tarau 2004; Thakkar et al. 2010) have been briefly described in the previous Sect. 2.

The prime concern of the proposed method is to find the best similarity measure for nodes of the graph. Here, we want to assign the weight to each edge of the graph in a manner such that, it incorporates a numeric factor which is the quantification of similarity with the overall document (specifically topic embedded in the document). We need to ensure that, once an edge is formed between two nodes (i.e., sentences), the weight assigned should be reliant on following two constraints

1. How much similar (statistically or semantically) those nodes are and
2. How much those two nodes are similar to the overall document (or overall document theme that is the topic representation of the input document)

**Fig. 2** General steps followed by the proposed method



To find the similarity between two sentences, i.e.,  $S_i$  and  $S_j$  and assigning the weight to the edges, we introduce the formula as

$$W(S_i, S_j) = A * Sim(S_i, S_j) + (1 - A) * Sim(S_{ij}, O_D)$$

where the terms used in the proposed formula are characterized as below

■  $W(S_i, S_j)$ :

It is the weight assigned to the edge connecting nodes (i.e., sentences)  $S_i$  and  $S_j$ .

■  $S_{ij}$ :

It is used to denote the vector formed with the word of both the sentences  $S_i$  and  $S_j$ , i.e., the union of the words that are contained in sentences  $S_i$  and  $S_j$ .

■  $O_D$ :

$O_D$  is the set of words that represent the overall (input) document in a smaller dimension provided that the theme of the input document is preserved. The primary goal of introducing  $O_D$  is to calculate the similarity of nodes to the input document. As we know that length of the sentences is smaller than of the input document. We represent input document in a smaller dimension (i.e.,  $O_D$ ) because we want to maintain dimensionality proportion while calculating similarity among the nodes and similarity among the nodes and the overall document.

Here, the primary concern is the manner by which the input document is represented in a smaller dimension i.e.,  $O_D$ . We need a mechanism that creates a subset of input text containing major topics of the original document. As we know, the keywords are the words or group of words with the help of which we get the ideas and topics that define what the content of the input document is about. So one of the



most straightforward techniques to find the theme of the text document is to represent it in the form of keywords. Topic modeling is a good alternative to represent the document in the forms of topic and respective words distribution. To generate  $O_D$ , we use LDA (Blei et al. 2003) topic modeling methodology. LDA is a generative probabilistic model that is applied on a text corpus. It generates the output as the topics with associated words and the respective probabilities. We apply LDA on input document and include top  $W_T$  words of the  $T$  topics in  $O_D$ .

Therefore, to generate the overall representation of the input document in reduced dimension i.e.,  $O_D$  we have used top  $W_T$  words (according to probabilities) from each of the  $T$  topics, where  $W_T$  and  $T$  are decided by the user.

■  $A$  :

$A$  is a factor which decides how much weight is assigned to the similarity values between node, equivalently  $(1 - A)$  is the weight assigned to the similarity between  $S_{ij}$  and  $O_D$ .

This parameter is introduced to assign the weight to the following two factors (1) How much similar edge forming nodes are and (2) How much edge forming nodes are syntactically or semantically close to the overall document (or document theme that is the topic representation of the input document, i.e.,  $O_D$ ). The user can decide the value of the parameter.

■  $Sim(S_i, S_j)$ :

It is the similarity measure between the sentences  $S_i$  and  $S_j$ . According to available literature, there is a large number of similarity measures available for finding the text similarity. The most fundamentally used similarity measures are Cosine measure, Jaccard similarity and Euclidean similarity, etc. In addition, various similarity measures have been proposed for graph-based text summarization. These measures (graph-based) have been described in the previous section briefly. Semantic similarity between the sentences can also be a better alternative for calculating  $Sim(S_i, S_j)$ . One can create word vectors of the input sentences and use one of the following measures to determine the similarity between the sentences i.e.,  $S_i$  and  $S_j$

□ Cosine measure:

$Cosine\ Similarity = \frac{\sum_{k=1}^N S_{ik} S_{jk}}{\sqrt{\sum_{k=1}^N S_{ik}^2} \sqrt{\sum_{k=1}^N S_{jk}^2}}$  (where  $S_{ik}$  and  $S_{jk}$  are the components of vector  $S_i$  and  $S_j$  respectively).

□ Euclidean distance:

$$Euclidean\ Distance = \sqrt{\sum_{k=1}^N |S_{ik} - S_{jk}|^2}$$

□ Jaccard Measure:

$Jaccard\ Coefficient = \frac{|W_{S_i} \cap W_{S_j}|}{|W_{S_i} \cup W_{S_j}|}$  (where  $W_{S_i}$  and  $W_{S_j}$  are the words in the sentence  $S_i$  and  $S_j$  respectively)

□ Learning based methods

Word embeddings based trained models are used to predict the similarity between the sentences i.e. Word2Vec or Doc2vec (Le and Mikolov 2014; Mikolov et al. 2013a, b)

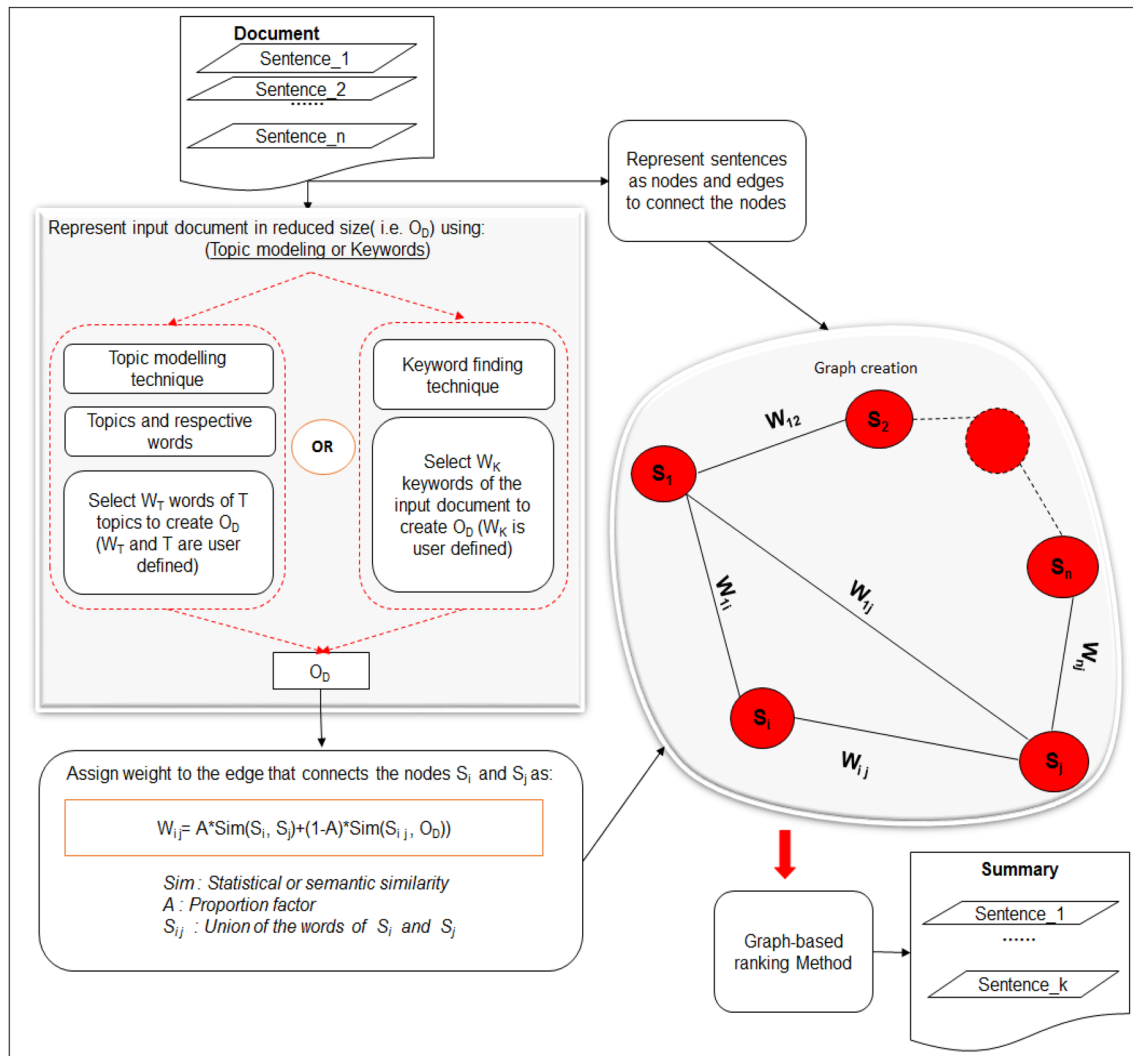
□ Methods used in graph-based techniques

The measure used in different graph-based text summarization methods (Barrios et al. 2016; Erkan and Radev 2004; Mihalcea 2004; Mihalcea and Tarau 2004). These methods are discussed in detail in the previous Section 2.

□ Semantic similarity

These are the methods which aimed to incorporate the semantic meaning for the sentence similarity. WordNet groups English words into synsets (sets of synonyms) and records the relations among these synsets (Miller 1995). The similarity between sentence  $S_i$  and  $S_j$  is computed on the basis of semantic similarity between each of the words in  $S_i$ , to each of the words in  $S_j$  using a distance measure based on WordNet. WordNet is extensively used lexical database for the English language (Miller 1995). It was created in the Princeton University under the direction of professor George Armitage Miller.

In the proposed method, we make the use of WordNet for the similarity calculation. The reason for using the WordNet



**Fig. 3** Proposed method: creation of graph and summary generation

is to incorporate the semantic meaning for the similarity calculation.

Once the graph is created, and weight is assigned to the edges using the proposed formula (as discussed earlier), we apply the graph-ranking algorithm to find the relevance of the sentences. Here we apply the weighted PageRank (Brin and Page 1998) algorithm as used in TextRank (Mihalcea and Tarau 2004; Mihalcea 2004).

Figure 3 shows how the proposed methodology creates the graph, and weights are assigned to the edges of the graph. Once the graph is created and weights are assigned to the edges using the proposed formula, the graph ranking algorithms can be applied to find the ranks of the sentences. Algorithm 1 describes the detailed steps followed by the proposed method.

**Algorithm 1** A New Graph-based Extractive Text Summarization Using Topic Modeling**Input**

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

$K$ : The number of sentences to be included in the summary (user-defined).

$A$ : A factor to decide the proportion of similarity measure (user-defined).

$W_k$ : The number of keywords to be selected from the input document (user-defined).

$T$ : The numbers of topics to be selected from the input document (user-defined).

$W_T$ : The number of words to be selected from each topic (user-defined).

**Output**

$S_O$ : The extractive summary of the given document. Where  $S_O \subset S$

**Steps**

**1:** Decompose the given document into the set of sentences.

**2:** Preprocess the text (i.e. stop word removal, punctuation removal, lemmatization, etc).

**3:** Create the graph using input sentences as following

**3.1:** Represent sentences as nodes of the graph.

**3.2:** Connect the nodes to find the edges of the graph.

**3.3:** Assign the weight  $W_{ij}$  to the edges (i.e. between  $S_i$  and  $S_j$ ) as follows

**3.3.1:** Create the vector  $S_{ij}$  (formed by the union of the words of sentences  $S_i$  and  $S_j$ ).

**3.3.2:** Create the vector  $O_D$  (top  $W_k$  keywords of the input document).  
OR

**3.3.2:** Create the vector  $O_D$  (top  $W_T$  words of  $T$  topic using LDA on input document).

**3.3.3**  $W(i, j) = A * \text{Similarity between } S_i \text{ and } S_j + (1-A) * \text{Similarity between } S_{ij} \text{ and } O_D$ .

**4:** Apply the graph ranking algorithm on the graph to find the rank of nodes (i.e. sentences).

**5:** Select the top  $K$  (according to the rank decided in the above step) sentences as a summary.

Figure 4 shows the example text containing eight sentences. After applying the proposed formula, we find the graph with associated edges weights, as given in Fig. 5.

Once the graph has been generated for the input text, graph-based ranking techniques can be applied to find the rank of the nodes (sentences). Suppose we want to generate three sentences summary for the text given in Fig. 4. After applying the PageRank algorithm in the graph given in Fig. 5, the nodes  $S_5, S_6, S_7$  have been assigned as first, second and third highest ranks respectively.

## 4 Evaluation

The proposed method is evaluated on two publicly available benchmark datasets. We have compared the proposed method with various techniques of text summarization.

### 4.1 Dataset and evaluation parameters

We have used two datasets for the evaluation of the proposed method. The Opinosis dataset (Ganesan et al. 2010) is comprised of 51 documents that contain the sentences gathered

Fig. 4 Example (input text)

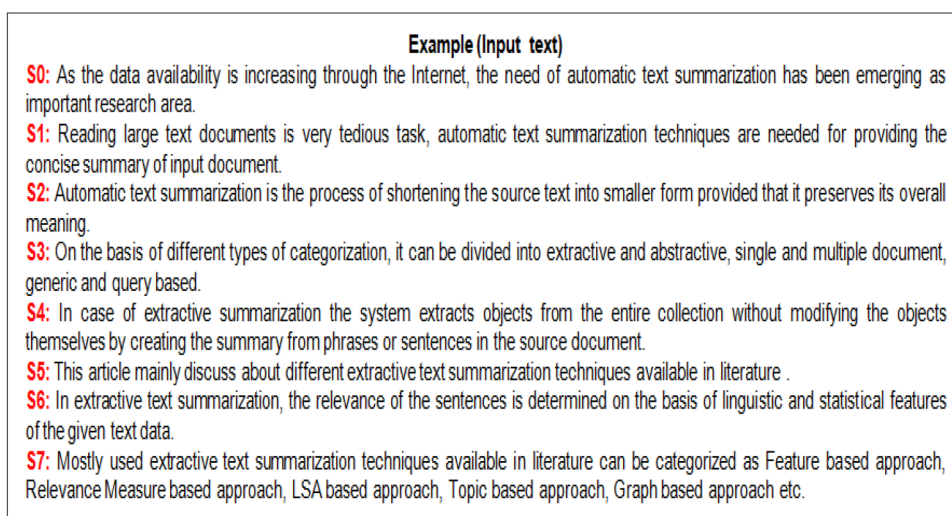
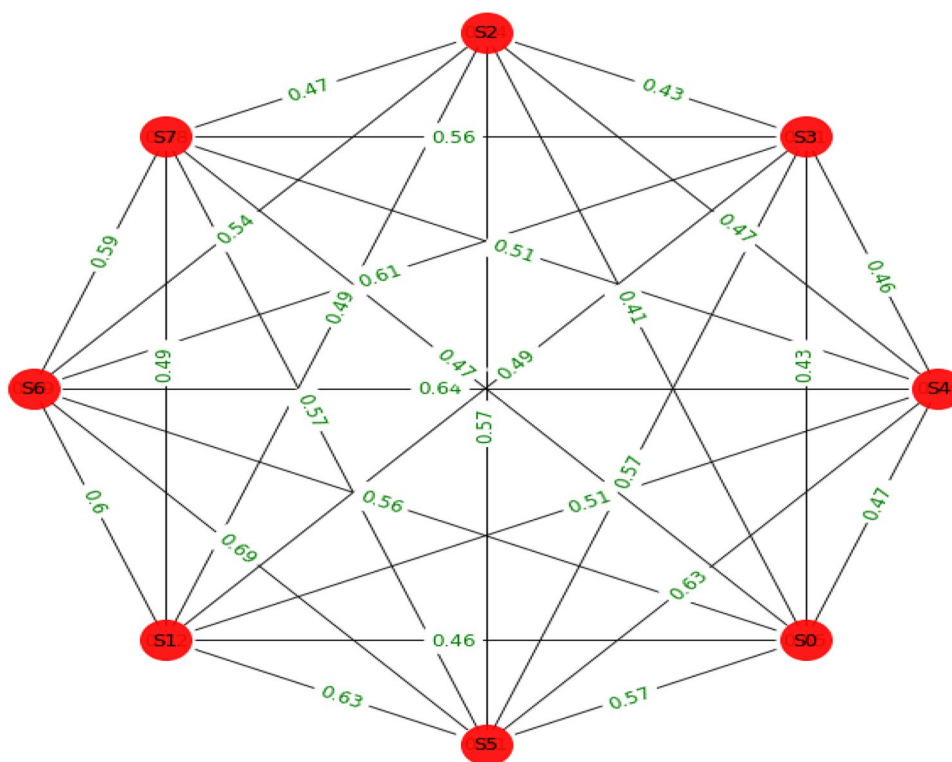


Fig. 5 Graph generated for the input text (given in Fig. 4)



from user reviews on the given topic collected from the different websites (Edmunds.com (cars), Amazon.com (various electronics) and Tripadvisor (hotels)). The dataset (Ganesan et al. 2010) contains 51 topics (each has approximately 100 sentences per topic) along with 5 gold summaries for each of the documents. The second dataset is CNN/ DailyMail (Hermann et al. 2015) comprises of different stories (related to political, social, and technical articles) collected from CNN/DailyMail news. In CNN/DailyMail dataset each of

the articles contains the respective highlights which are used as the summary for the article.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin 2004) is a method used for the evaluation of automatic summarization. The mainly used ROUGE metric for the evaluation are

**ROUGE-N:** is used to evaluate the overlap of N-grams between the system and reference summaries.

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

Methods	R-1	R-2	R-L
RM (Gong and Liu 2001)	0.355	0.130	0.173
LUHN (Luhn 1958)	0.364	0.130	0.169
TextRank (Mihalcea 2004)	0.383	0.145	0.196
LexRank (Erkan and Radev 2004)	0.385	0.140	0.207
LSA (Ozsoy et al. 2011)	0.344	0.122	0.170
TextRank Variation (Barrios et al. 2016)	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 et al. 2018b)	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
PGAN-ATSMT (Yang et al. 2020)	0.421	0.199	0.389
Proposed method	0.428	0.201	0.392

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

Methods	R-1	R-2	R-L
RM (Gong and Liu 2001)	0.153	0.031	0.059
LUHN (Luhn 1958)	0.154	0.032	0.053
TextRank (Mihalcea 2004)	0.193	0.044	0.075
LexRank (Erkan and Radev 2004)	0.269	0.083	0.152
LSA (Ozsoy et al. 2011)	0.167	0.033	0.066
TextRank Variation (Barrios et al. 2016)	0.177	0.040	0.073
Proposed Method	0.271	0.084	0.161

**ROUGE-1:** is used to evaluate the overlap of 1-gram (every single word) between the system and reference summaries.

**ROUGE-2:** is used to evaluate the overlap of bi-grams between the system and reference summaries.

**ROUGE-L:** is used to evaluate the Longest Common Subsequence between the system and reference summaries.

In addition, the symbols P, R, and F that are associated with ROUGE are used to refer to Precision, Recall, and F-measure, respectively, and can be defined as below Mirshojaee et al. (2020)

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

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

$$Fscore = \frac{2 * Precision * Recall}{Precision + Recall}$$

**Table 3** Comparison among the different text summarization methods for maximum value of Precision (Opinosis Dataset)

Methods	R-1	R-2	R-L
RM (Gong and Liu 2001)	0.094	0.018	0.057
LUHN (Luhn 1958)	0.095	0.018	0.051
TextRank (Mihalcea 2004)	0.129	0.027	0.071
LexRank (Erkan and Radev 2004)	0.222	0.061	0.143
LSA (Ozsoy et al. 2011)	0.108	0.019	0.063
TextRank Variation (Barrios et al. 2016)	0.116	0.023	0.070
Proposed method	0.178	0.062	0.144

**Table 4** Comparison among the different text summarization methods for maximum value of Recall (Opinosis Dataset)

Methods	R-1	R-2	R-L
RM (Gong and Liu 2001)	0.553	0.161	0.455
LUHN (Luhn 1958)	0.549	0.180	0.451
TextRank (Mihalcea 2004)	0.521	0.170	0.414
LexRank (Erkan and Radev 2004)	0.476	0.186	0.396
LSA (Ozsoy et al. 2011)	0.539	0.149	0.415
TextRank Variation (Barrios et al. 2016)	0.545	0.178	0.452
Proposed method	0.563	0.187	0.460

**Table 5** 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 and Liu 2001)	0.104	0.062	0.413
LUHN (Luhn 1958)	0.103	0.060	0.420
TextRank (Mihalcea 2004)	0.133	0.085	0.382
LexRank (Erkan and Radev 2004)	0.190	0.148	0.331
LSA (Ozsoy et al. 2011)	0.112	0.069	0.387
TextRank Variation (Barrios et al. 2016)	0.122	0.075	0.392
Gist( Lovinger et al. 2019)	0.220	0.153	0.421
Proposed method	0.229	0.154	0.445

We have discussed various measures ( $Sim(S_i, S_j)$ ) to be used as a similarity value. After the preprocessing step is applied, and the input text is transformed into a vector form, any similarity method can be used to assign a weight to the graph edges. We have discussed the novelty of semantic meaning in the context of the proposed method; as a result, we have applied semantic measures while evaluating the results (discussed in the previous section).

The value of A may vary between 0 and 1. Assigning 1 to A means, we give full weight to the similarity among the nodes, that are forming the edges (like existing methods of text summarization). While assigning 0 to A means, we

provide full weight to the similarity between edge forming nodes and overall document theme, i.e.,  $O_D$ . To calculate edges weight, i.e.,  $W$ , we have taken the value of parameter  $A$  as 0.5. For the assessment of the proposed technique, we have taken the number of topic words, i.e.,  $W_T$  as 10% (of the given text's total words). Moreover, we have taken number of topics, i.e.,  $T$  equal to the number of sentences to be included in the summary.

## 4.2 Results

To evaluate the effectiveness of the proposed method, we have compared the results with different text summarization methods. Those methods are RM method (Gong and Liu 2001), Luhn method (Luhn 1958), TextRank method (Mihalcea 2004), LSA method (Ozsoy et al. 2011), TextRank Variation method (Barrios et al. 2016), Sumrunner (Nallapati et al. 2017), REFRESH (Narayan et al. 2018), LATENT (Zhang et al. 2018b), SUMO (Liu et al. 2019), VHTM (Fu et al. 2020), Gist (Lovinger et al. 2019) and PGAN-ATSMT (Yang et al. 2020) method.

Evaluation results on two benchmark datasets demonstrate that the proposed method achieves better performance than the existing methods in terms of ROUGE matrices.

Table 1 represents the comparison among the various methods of the text summarization, which are evaluated on CNN/DailyMail dataset for F-measure. Here R-1, R-2, R-L represent the Rouge-1 (unigram or 1gram), Rouge-2 (bigram) and Rouge - L (Longest common subsequence) respectively. Table 2 represents the comparison among the various methods, calculating the maximum F-measure value (out of five gold summaries for a particular text) evaluated on Opinions dataset.

Table 3 demonstrates a comparison among various approaches, measuring the maximum precision value (out of five gold summaries for a given text) evaluated on the Opinions dataset. Table 4 is the comparison among the various methods, figuring out the maximum (out of five gold summaries for a particular text) Recall value evaluated on Opinions dataset. Table 5 represents the comparison of the average value of F-measure, Precision and Recall value evaluated on Opinions dataset. Here average word stands for the average value out of the five gold summaries.

All of the above methods are evaluated on CNN/ DailyMail (Hermann et al. 2015) dataset. In Opinions (Ganesan et al. 2010), seven methods are evaluated because the remaining methods are based on learning techniques, need a large amount of the data to learn the parameters. CNN/ DailyMail dataset have a sufficient number of the instances (i.e., input text and respective summary). On the other hand, Opinions dataset has 51 documents (with five gold summaries for each document). Moreover, this dataset (Opinions) is sufficient for unsupervised techniques.

The proposed method is compared with thirteen state-of-the-art text summarization approaches, belonging to various categories. i.e., feature-based, relevance measure based, learning-based (both supervised and unsupervised), graph-based, topic-based, tree-based, and LSA-based. These methods have been compared with respect to F-measure, Precision, and Recall for ROUGE-1, ROUGE-2, and ROUGE-L tools. For the Opinions (Ganesan et al. 2010) dataset, the values are compared for maximum and average values (out of five gold summaries) for the ROUGE scores. The evaluation results can be easily observed; ROUGE values obtained demonstrate the out-performance of the proposed method over existing text summarization methods.

## 5 Conclusion and future work

It has been observed from the proposed technique while calculating the similarity score between the nodes, the numeric factor, which represents the similarity between the edges forming nodes and overall document, also plays an important role in the rank calculation. In graph-based summarization, for calculating the weight of the edges, most of the literature is specific on the similarity measure among the nodes. Whereas, we have incorporated additional parameter which calculates the similarity of the nodes to the overall document too. We have incorporated the topic modeling to overcome the problem of redundancy associated with existing methods of summarizations.

Results obtained from the evaluation on two datasets CNN/DailyMail (Hermann et al. 2015) and Opinions (Ganesan et al. 2010) demonstrate the outperformance of the proposed method. The summary generated by the proposed method scores better F-measure while compared with existing extractive text summarization techniques.

In the future, we would like to extend the proposed idea for graph clustering based text summarization techniques.

## References

- Abdi A, Shamsuddin SM, Aliguliyev RM (2018a) Qmos: Query-based multi-documents opinion-oriented summarization. *Inform Process Manage* 54(2):318–338
- Abdi A, Shamsuddin SM, Hasan S, Piran J (2018b) Machine learning-based multi-documents sentiment-oriented summarization using linguistic treatment. *Expert Syst Appl* 109:66–85. <https://doi.org/10.1016/j.eswa.2018.05.010>
- Ali SM, Noorian Z, Bagheri E, Ding C, Al-Obeidat F (2020) Topic and sentiment aware microblog summarization for twitter. *J Intell Inform Syst* 54(1):129–156. <https://doi.org/10.1007/s1084-4-018-0521-8>
- Allahyari M, Pouriyeh S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, Kochut K (2017) Text summarization techniques: a brief survey. *arXiv preprint arXiv:170702268*

- Alterman R (1991) Understanding and summarization. *Artif Intell Rev* 5(4):239–254
- Amplayo RK, Song M (2017) An adaptable fine-grained sentiment analysis for summarization of multiple short online reviews. *Data Knowl Eng* 110:54–67
- Arora R, Ravindran B (2008) Latent dirichlet allocation based multi-document summarization. In: Proceedings of the second workshop on Analytics for noisy unstructured text data. ACM, pp 91–97
- Barrios F, López F, Argerich L, Wachenchauer R (2016) Variations of the similarity function of textrank for automated summarization. arXiv preprint [arXiv:160203606](https://arxiv.org/abs/160203606)
- Barros C, Lloret E, Saquete E, Navarro-Colorado B (2019) Natsum: Narrative abstractive summarization through cross-document timeline generation. *Inform Process Manag* 56(5):1775–1793
- Baxendale PB (1958) Machine-made index for technical literature—an experiment. *IBM J Res Dev* 2(4):354–361. <https://doi.org/10.1147/rd.24.0354>
- Bellaachia A, Al-Dhelaan M (2012) Ne-rank: A novel graph-based keyphrase extraction in twitter. In: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, IEEE, vol 1, pp 372–379
- Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. *J Mach Learn Res* 3:993–1022
- Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. *Comput Netw ISDN Syst* 30(1–7):107–117
- Chang YL, Chien JT (2009) Latent dirichlet learning for document summarization. In: 2009 IEEE international conference on acoustics, speech and signal processing, IEEE, pp 1689–1692
- Cuong HN, Tran VD, Van LN, Than K (2019) Eliminating overfitting of probabilistic topic models on short and noisy text: the role of dropout. *Int J Approx Reason*
- Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. *J Am Soc Inform Sci* 41(6):391–407
- Edmundson HP (1969) New methods in automatic extracting. *J ACM (JACM)* 16(2):264–285. <https://doi.org/10.1145/321510.321519>
- Erkan G, Radev DR (2004) Lexrank: Graph-based lexical centrality as salience in text summarization. *J Artif Intell Res* 22:457–479
- Fattah MA (2014) A hybrid machine learning model for multi-document summarization. *Appl Intell* 40(4):592–600
- Fattah MA, Ren F (2008) Automatic text summarization. *World Acad Sci Eng Technol* 37:2008
- Ferreira R, de Souza CL, Lins RD, e Silva GP, Freitas F, Cavalcanti GD, Lima R, Simske SJ, Favaro L, (2013) Assessing sentence scoring techniques for extractive text summarization. *Expert Syst Appl* 40(14):5755–5764
- Fuad TA, Nayeem MT, Mahmud A, Chali Y (2019) Neural sentence fusion for diversity driven abstractive multi-document summarization. *Comput Speech Language* 58:216–230
- 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
- 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 (Coling 2010), pp 340–348
- 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, ACM, pp 19–25
- Gupta P, Pendluri VS, Vats I (2011) Summarizing text by ranking text units according to shallow linguistic features. In: 13th International Conference on Advanced Communication Technology (ICACT2011), IEEE, pp 1620–1625
- Haiduc S, Aponte J, Moreno L, Marcus A (2010) On the use of automated text summarization techniques for summarizing source code. In: 2010 17th Working Conference on Reverse Engineering, IEEE, pp 35–44
- Harabagiu SM, Lacatusu VF, Morarescu P (2002) Multidocument summarization with gistexter. *LREC Citeseer* 1:1456–1463
- Herings P, Van der Laan G, Talman D (2001) Measuring the power of nodes in digraphs. Gerard and Talman, Dolf JJ, Measuring the Power of Nodes in Digraphs (October 5, 2001)
- Hermann KM, Kocisky T, Grefenstette E, Espeholt L, Kay W, Suleyman M, Blunsom P (2015) Teaching machines to read and comprehend. In: Advances in neural information processing systems, pp 1693–1701
- Iyer S, Konstas I, Cheung A, Zettlemoyer L (2016) Summarizing source code using a neural attention model. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol 1, pp 2073–2083
- Kleinberg JM (1999) Authoritative sources in a hyperlinked environment. *J ACM (JACM)* 46(5):604–632
- Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on machine learning, pp 1188–1196
- Li X, Wang Y, Zhang A, Li C, Chi J, Ouyang J (2018) Filtering out the noise in short text topic modeling. *Inf Sci* 456:83–96
- Lim KW, Buntine W, Chen C, Du L (2016) Nonparametric bayesian topic modelling with the hierarchical pitman-yor processes. *Int J Approx Reason* 78:172–191
- Lin CY (2004) Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*
- 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. *International Conference on Text, Springer, Speech and Dialogue*, pp 16–23
- Lovinger J, Valova I, Clough C (2019) Gist: general integrated summarization of text and reviews. *Soft Comput* 23(5):1589–1601
- Luhn HP (1958) The automatic creation of literature abstracts. *IBM J Res Dev* 2(2):159–165
- 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 Syst Appl* 133:173–181
- Mihalcea R (2004) Graph-based ranking algorithms for sentence extraction, applied to text summarization. In: Proceedings of the ACL Interactive Poster and Demonstration Sessions
- Mihalcea R, Tarau P (2004) Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing
- Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. arXiv preprint [arXiv:1301.781](https://arxiv.org/abs/1301.781)
- Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013b) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
- Miller GA (1995) Wordnet: a lexical database for english. *Commun ACM* 38(11):39–41
- Mirshojaee SH, Masoumi B, Zeinali E (2020) Mamhoa: a multi-agent meta-heuristic optimization algorithm with an approach for document summarization issues. *J Ambient Intell Humaniz Comput* 1–16
- Mutlu B, Sezer EA, Akcayol MA (2019) Multi-document extractive text summarization: a comparative assessment on features. *Knowl-Based Syst* 183:104848

- 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
- 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 (2014 CCDC), IEEE, pp 5168–5172
- Narayan S, Cohen SB, 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
- Narayan S, Papasrantopoulos N, Cohen SB, Lapata M (2017) Neural extractive summarization with side information. arXiv preprint [arXiv:170404530](https://arxiv.org/abs/170404530)
- Nenkova A, McKeown K (2012) A survey of text summarization techniques. In: Mining text data, Springer, pp 43–76
- Nguyen MT, Tran VC, Nguyen XH, Nguyen LM (2019) Web document summarization by exploiting social context with matrix cofactorization. *Inform Process Manag* 56(3):495–515
- Nguyen-Hoang TA, Nguyen K, Tran QV (2012) Tsgvi: a graph-based summarization system for vietnamese documents. *J Ambient Intell Humaniz Comput* 3(4):305–313
- Ouyang Y, Li W, Li S, Lu Q (2011) Applying regression models to query-focused multi-document summarization. *Inform Process Manag* 47(2):227–237
- Ozsoy MG, Alpaslan FN, Cicekli I (2011) Text summarization using latent semantic analysis. *J Inform Sci* 37(4):405–417
- Rush AM, Chopra S, Weston J (2015) A neural attention model for abstractive sentence summarization. arXiv preprint [arXiv:150900685](https://arxiv.org/abs/150900685)
- Saggion H, Poibeau T (2013) Automatic text summarization: Past, present and future. In: Multi-source, multilingual information extraction and summarization, Springer, pp 3–21
- Thakkar KS, Dharaskar RV, Chandak M (2010) Graph-based algorithms for text summarization. In: 2010 3rd International Conference on Emerging Trends in Engineering and Technology, IEEE, pp 516–519
- Van Lierde H, Chow TW (2019) Query-oriented text summarization based on hypergraph transversals. *Inform Process Manag* 56(4):1317–1338
- Vetriselvi T, Gopalan N (2020) An improved key term weightage algorithm for text summarization using local context information and fuzzy graph sentence score. *J Ambient Intell Humaniz Comput* 1–10
- Xu GX, Yao HS, Wang C (2017) Research on multi-feature fusion algorithm for subject words extraction and summary generation of text. *Cluster Comput* 1–13
- 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. *Inf Sci* 521:46–61
- Yousefi-Azar M, Hamey L (2017) Text summarization using unsupervised deep learning. *Expert Syst Appl* 68:93–105
- Zhang L, Wu Z, Bu Z, Jiang Y, Cao J (2018a) A pattern-based topic detection and analysis system on Chinese tweets. *J Comput Sci* 28:369–381
- Zhang X, Lapata M, Wei F, Zhou M (2018b) Neural latent extractive document summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp 779–784

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.