Research Article

Manju Kondath*, David Peter Suseelan, and Sumam Mary Idicula

Extractive summarization of Malayalam documents using latent Dirichlet allocation: An experience

https://doi.org/10.1515/jisys-2022-0027
received July 07, 2021; accepted February 26, 2022

Abstract: Automatic text summarization (ATS) extracts information from a source text and presents it to the user in a condensed form while preserving its primary content. Many text summarization approaches have been investigated in the literature for highly resourced languages. At the same time, ATS is a complicated and challenging task for under-resourced languages like Malayalam. The lack of a standard corpus and enough processing tools are challenges when it comes to language processing. In the absence of a standard corpus, we have developed a dataset consisting of Malayalam news articles. This article proposes an extractive topic modeling-based multi-document text summarization approach for Malayalam news documents. We first cluster the contents based on latent topics identified using the latent Dirichlet allocation topic modeling technique. Then by adopting vector space model, the topic vector and sentence vector of the given document are generated. According to the relevant status value, sentences are ranked between the document's topic and sentence vectors. The summary obtained is optimized for non-redundancy. Evaluation results on Malayalam news articles show that the summary generated by the proposed method is closer to the human-generated summaries than the existing text summarization methods.

Keywords: topic modeling, LDA, multi-document summarization, relevant status value, Malayalam documents

1 Introduction

In today's digital world, we have different websites providing news in large quantities. Many of these websites provide the same news by changing some details, and most do not provide complete information to the reader. If people read more than one news article about the same topic, there is a high probability of redundancy in the data. It will be advantageous if all the information from the different sources is summarized and presented to the reader in a condensed form. This can be achieved with multi-document summarization systems. These systems aim to create abstracts from multiple documents by selecting sentences with premium content and discarding redundant information.

Automatic text summarization (ATS) extracts relevant data from one or more text documents to create a condensed version of the original document(s). Many approaches to document summarization have been suggested. Text summarization is characterized as either single document or multi-document based on the number of source documents used. Based on the way the final summary is generated, the summarization

* **Corresponding author: Manju Kondath,** Department of Computer Science, Cochin University of Science and Technology, Kochi 682022, Kerala, India, e-mail: manju@mec.ac.in
David Peter Suseelan: Department of Computer Science, SOE Campus, Cochin University of Science and Technology, Kochi 682022, Kerala, India, e-mail: davidpeter@cusat.ac.in
Sumam Mary Idicula: Department of Computer Science, Muthoot Institute of Technology and Science, Kochi 682308, Kerala, India, e-mail: sumamdavid123@gmail.com

methods are broadly defined as extractive and abstractive summarization [1] methods. The extractive summarization method creates a summary by extracting significant sentences from a collection of documents without modifying those sentences. In general, the information content is not distributed evenly across each sentence. As a result, it is efficient to find the subset of sentences that serves as the document summary [2]. On the other hand, the abstractive summarization method produces a summary by automatically rephrasing the sentences that carry the most relevant information or generating new sentences with concepts residing in the document.

The majority of extractive multi-document summary systems use unsupervised or supervised learning techniques to determine the importance of sentences. If the data are labeled, we can use supervised or semi-supervised machine learning to train a classifier that predicts the significance of a sentence [3]. While supervised and semi-supervised learning can produce excellent results, many datasets are unlabeled, thus ruling out these methods. It is possible to label the data manually; however, the volume of labeled training data required makes this approach difficult. The same holds for a semi-supervised strategy that requires less labeled data. As a result, extractive summarization frequently relies on unsupervised learning [3].

In unsupervised learning, feature-based ranking methods have been widely used. Feature-based techniques use various linguistic and statistical features to determine the relevance of sentences in summary [1]. Topic-based approaches are based on the distribution of topic words in the input text, and modeling techniques have been integrated into the summarization method to find the summary. In practice, each sentence within the given text represents a topic embedded in the document. One of the strategies for defining the contextual content of text documents has been topic recognition. Latent topics can be used to measure correlations between documents [4]. Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus [5,6]. LDA has been used successfully in multi-document summarization [7–11].

ATS faces a variety of challenges that hones its development. Apart from content selection, redundancy in the text is a critical issue that should be addressed to generate a perfect summary. This problem becomes more apparent in multi-document summarization. The readability of the summary is the next challenge in multi-document summarization. This means that the approach used must organize the selected sentences so that the generated summary has coherence, resulting in appropriate readability. Other difficulties stem from the nature of the language being processed. Different languages have their own grammatical and morphological structures, semantic rules, and lexicon resulting in a text that is difficult to process. Language issues pose numerous challenges during the preprocessing phase of summarization.

Several forms of automated document summarizers for English [12] and other foreign languages, such as Arabic [13,14], Chinese [15,16], and French [17], have been addressed. Some ATS systems have also been developed for Indian languages such as Gujarati [18], Punjabi [19], Urdu [20], Marathi [21], and Tamil [22]. Malayalam is a regional language of India spoken by the people of Kerala. There are numerous Malayalam online newspapers available. Because of the excessive overloading of data on the internet, ATS applications are needed to address this problem. Since Malayalam lacks a multi-document summarizer, an ATS is proposed. Malayalam research [23] is still in its early stages and there is an increasing need to build systems for processing and summarizing Malayalam texts. Language processing in this language is difficult due to its highly agglutinated and inflectional features.

Inspired by the success of LDA, we propose an unsupervised extractive summarization approach that selects only the sentences that contain most topic terms embedded in them. The generated summary contains sentences that represent the main content. The proposed method uses three significant steps for generating the summary. The first step is to use LDA to generate the topic vector for the given document. The sentences in the text are then vectorized in relation to the topic vectors. Subsequently, find the relevance of each sentence based on the similarity with the topic vector. The last step is to rank the sentences based on their relevance so that top-ranked sentences are included in the summary.

Experiments were carried out using a dataset explicitly created for Malayalam ATS. The efficiency of this approach is demonstrated by an automated evaluation of these data sets.

To summarize, the main contributions of this work are as follows:

- We propose an unsupervised method based on topic modeling and MMR for extractive multi-document summarization of Malayalam documents, aiming to provide maximum coverage of the presented content and minimum redundancy.
- Using the redundancy removal component, we could eliminate similar sentences and diversify the information in the final summary of the document.
- Malayalam lacks a benchmark ATS dataset like the document understanding conference (DUC) dataset for English. As a result, we have created a multi-document ATS dataset for the Malayalam language in consultation with the language experts from Malayalam University.
- We compared the proposed work with the Textrank [11] model on the data set prepared for Malayalam ATS. We also compared the results with the results obtained for ATS developed for other Indian languages. The results show that the proposed method outperforms the baseline and yields comparable performance with works done for other languages.

The remainder of this article is organized as follows. Section 2 presents the related works in LDA. Section 3 presents the methodology followed. Section 4 details the performance evaluation. The conclusion is presented in Section 5.

2 Related works

This section reviews some existing works proposed in the literature of multi-document extractive text summarization. When the input consists of multiple closely related documents, multi-document summarization is preferred [24]. The system is made simple by joining all input documents into one document and then building the final summary. Alternatively, summarize each document individually and then merge and construct the final summary [25].

In 1958, one of the first attempts at ATS was made, with the goal of extracting relevant sentences based on word frequency. After that, several attempts have been made using various approaches to develop text summarization systems. However, the statistical approach, topic-based approach, graph-based approach, and machine learning approaches are the most commonly used approaches in the literature [18]. The statistical approach extracts the most salient sentences based on shallow textual features such as the sentence's resemblance to the title, sentence position, the presence of numerical data in the sentence, the company of proper nouns (named entities), and the TF-IDF (term frequency inverse document frequency). Each of the features mentioned above gives the words some weight. The scores are assigned to the sentences based on these weights, and the highest scoring sentences are chosen to generate the summary. This method is language-agnostic.

Topic-based approaches infer topics by observing how words are distributed across documents. LDA was the first topic model to solve multi-document summarization, and researchers are constantly improving it. LDA considers each document to be a collection of different "topics" and each topic to be a collection of different "words" [5]. LDA [6] was proposed by David and his co-researchers in 2003 as a document representation method. It creates a topic per document model and a word per topic model based on Dirichlet distributions. They applied the LDA technique to evaluate the document model and found that LDA outperformed other latent topic models. The LDA-based Multi-document Summarization algorithm was proposed by Arora and Ravindran in 2008 [7]. Since the technique for creating topics is absolutely mathematical, and because it is based on probability distribution, we can evaluate this model in any language [26]. When applied to complex input text data such as legal documents [27], this technique proved very efficient.

Many ATS systems apply the graph-based methods for extractive text summarization such as LexRank [28], TextRank [29], and TextRankExt [30]. Graph-based approaches have achieved robust and promising results [31].

Deep-learning-based methods are used in some extractive MDS systems. Nallapati et al. proposed the SummaRuNNer model [32]. They used an RNN-based model to handle extractive summarization as a

sequence classification problem and used no attention mechanism. On the other hand, the hierarchical structured self-attentive extractive summarization model (HSSAS) [33] creates sentences and document embeddings using attention mechanisms in both the word and sentence layers. Each sentence is handled sequentially in the same order as the input document in SummaRuNNer and HSSAS. Then it is binary classified as to whether or not we should include it in the final summary. Some drawbacks of deep-learning-based systems include (1) the need for humans to manually build massive amounts of training data, and (2) the adaptation issues they may encounter when tested on a corpus other than the trained domain [33].

ROUGE [34] is the evaluation metric used for evaluating the approach concerning the dataset developed for the summarization task. Our method employs topic modeling to extract summaries from the given documents. To determine the importance of the sentence to be included in the description, we use the principle of LDA topic modeling in the proposed process. LDA is a generative probabilistic model applied to a discrete data collection, such as a text corpus. The model considers each word in the document to be an attribute of one of the document's topics. LDA represents each document as a random mixture of latent topics, and distribution over words characterizes each topic. The section that follows details the specific steps of the technique used.

3 Methodology

This section presents the dataset involved in the experiments and describes the process flow of the architecture. The architecture of the system is shown in Figure 1.

3.1 Dataset

As Malayalam document summarization is in a nascent stage, a standard dataset for summarization system is unavailable. So we have generated a dataset with 100 document sets, each with 3 news reports from 3 popular Malayalam e-newspapers: Mathrubhumi, Manorama, and Madhyamam, for the period March 2019 to October 2019. Table 1 displays the characteristics of the dataset. We collected 300 news articles on sports, politics, health, and entertainment from these three sources. Using web scraping tools, we picked articles related to the same topic from the sources and saved them as text files. The articles mostly ranged from 10 to 70 sentences. The next stage was to develop a manual summary to be used as a gold standard fitting to multi-document summarization. The PG students of the Computer Science Department who were fluent in Malayalam language prepared one reference summary for each group, for the automatic review of the system summaries. Due to the varying sizes of articles, it was impossible to predefine an exact number of words that would constitute the summary; therefore, we precalculated the number of candidate sentences to be equivalent to 40% of the most comprehensive article in the group. The dataset was validated by language experts from Malayalam University.[1]

The summarization task is intrinsically subjective; the length and content of the resulting summary differ between human evaluators. The content that should be included in the summary is frequently a point of contention among assessors [35]. The proportions of units chosen and non-chosen by the assessors and the ratio of shared decisions can be used to quantify the agreement between the two assessors on the summary content. Cohen's Kappa score was used to calculate the agreement amongst the human assessors [36]. We have also used ROUGE F1 scores to verify the reliability of the data set. ROUGE computes the overlap between two summaries in terms of n-grams recall.

1 https://github.com/Manju1974/Dataset

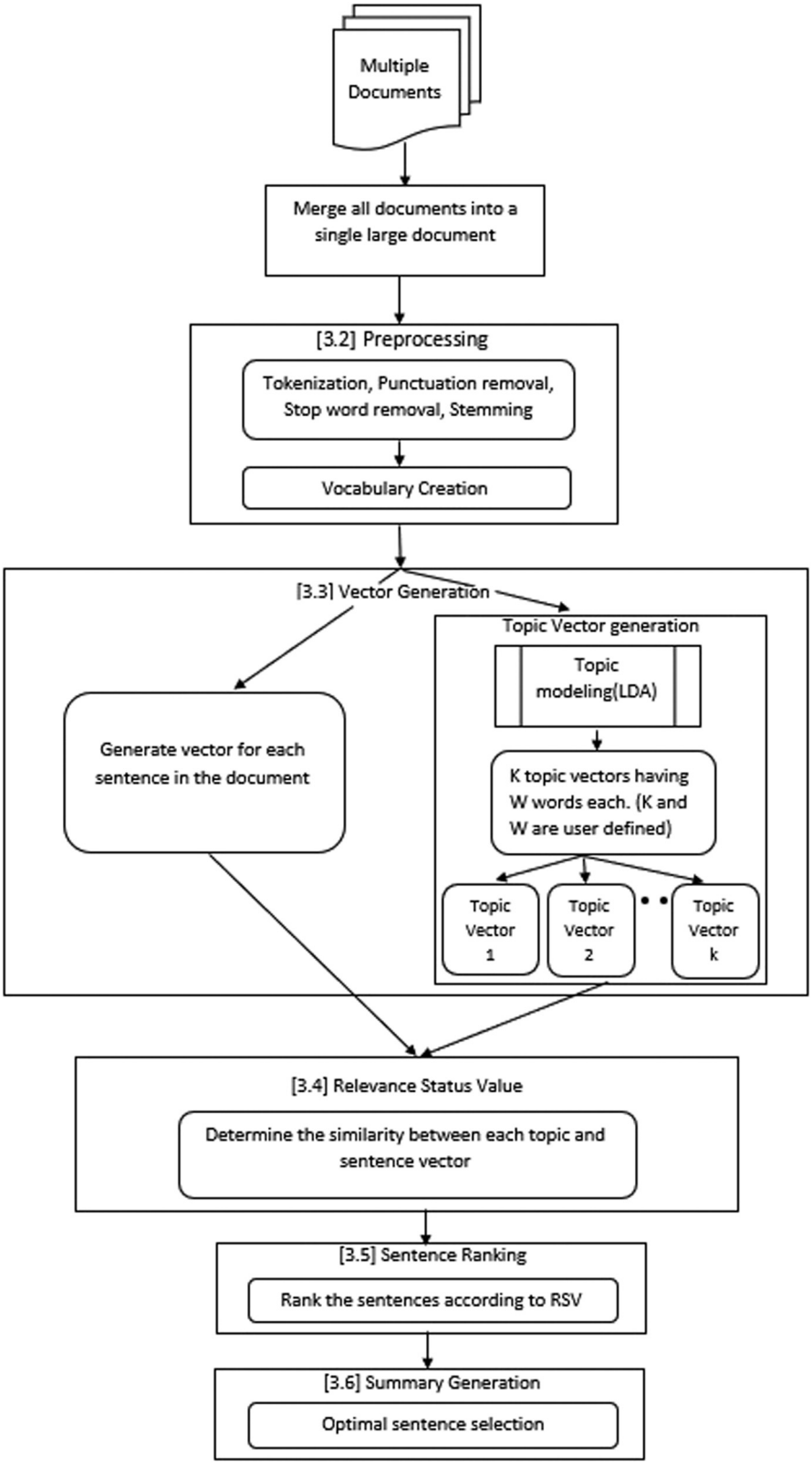

Figure 1: Methodology overview.

Table 1: Characteristic of the dataset

Dataset parameters	
Number of sets of documents	100
Number of documents in each set	3
Average number of sentences per document	21.7
Maximum number of sentences per document	70
Minimum number of sentences per document	10
Summary length (%)	40

The authenticity of the dataset was evaluated by taking the assistance of two language experts for summary generation. In order to compute the inter-annotator agreement between the human assessors we calculated the averaged Cohen's Kappa score. The Cohen's Kappa score obtained was 0.61, indicating that the annotators substantially agreed. In terms of the ROUGE-1 F1 score, the dataset received an average value of 0.77. All these results show that our dataset is trustworthy. It is important to note that our proposed approach is completely unsupervised because it does not use the actual summary information to generate the summary. The actual summary is used to evaluate our generated summary at the end of the execution of our algorithm.

To produce the extractive summary of the input documents, the proposed method employs the following steps.

3.2 Preprocessing

Text preprocessing is an unavoidable step in natural language processing. Furthermore, the accuracy of the preprocessing phase has a significant impact on the results of the main algorithm applied in the processing phase. The preprocessing step attempts to normalize the original text and adapt it to the format required for further processing. As a first step, aggregate the text of all documents in the input collection. Preprocessing involves sentence segmentation, tokenization, stopword elimination, and stemming. After aggregating the texts of document collection, the resultant document should be divided into constituent sentences. This is done during the sentence segmentation phase using Natural Language Toolkit module of Python. In the tokenization step, each sentence is split into tokens. Stop words are generally the most common words in a language that has little value in selecting the relevant sentences of an input document. Our system uses a stop word list with 85 words to filter out stop words. Removing stop words simplifies the vectorization of the sentence and improves the cosine similarity value. Stemming is an essential step in preparing input text since it reduces inflectional forms and occasionally derivational forms of a word to a common base form. This task has a favorable impact on improving the objective function's accuracy. In our proposed system, the stemmer used is similar to Indicstemmer [37], which follows an iterative suffix stripping algorithm to handle multiple levels of inflection. For example, while stemming, the Malayalam words *vanathiloode* and *vanathil* get transformed to the root form *vanam*. As a result, different forms of a word with the same root are treated as the same throughout the vector generation step of the processing phase. Thus, stemming is critical in improving the performances of the proposed method. We use Python gensim library to create dictionary (vocabulary) for the corpus.

3.3 Vector generation

The second step is to vectorize the input text and the topic words. Each sentence in the input text is represented as a vector. The most widely used vector representations are binary and TF-IDF [38]. Subsequently, we use topic modeling to generate topic vector. The LDA methodology is used to understand the

representation and distribution of topics in a text. Top W words are chosen from K topics based on the probability associated with the words, where W and K are chosen by the user. LDA learns the topic representation as follows:

(1) First, the number of topics to be discovered is determined (let it be K).
(2) Following that LDA will randomly allocate each of the words in each sentence to one of the K topics. As the words to topics are assigned at random the resulting representation is not optimal or accurate.
(3) LDA evaluates the percentage of words in the text assigned to a specific subject to improve the representation.
    $p(T|S)$ = the proportion of words in sentence S that are currently assigned to topic T.
    $p(W|T)$ = the proportion of all words W assigned to topic T in the sentence.
(4) Multiply $p(T|S)$ and $p(W|T)$ to get the conditional probability that the word takes on each topic. Reassign the word to the topic with the largest conditional probability.
(5) The preceding process is repeated for each word in each sentence in the text until convergence is reached.

The LDA generates K topics, each of which is a combination of keywords, with each keyword contributing a certain weightage to the topic.

An example can be used to describe the LDA topic vector generation process. Assume we have two documents from which to construct the summary, as shown in Figure 2. These two documents discuss about appointing Chief of Defence staff (CDS) for improving the coordination between defensive forces. Assume we need to produce three topics from the provided text. When we apply LDA modeling to the input text, we get the topic and the related terms, as shown in Figure 2. It is clear from this that the first topic pertains to a defensive upgrade, the second topic relates to the necessity for CDS, and the third topic comprises terminology related to the establishment of CDS.

3.4 Relevance status value (RSV)

The next step is to determine the relevance of the sentence vector with the topic vector. The relevance of a sentence with a topic is roughly equal to the sentence-topic vector similarity. Commonly used similarity measures are Cosine similarity, Jaccard similarity, and Euclidean measure. To determine the RSV we compute the cosine of the angle between the topic vector and the sentence vector as in equation (1).

$$\text{Cosine similarity } (T, S) = \frac{\sum_{k=1}^{n} T_i * S_i}{\sqrt{\sum_{k=1}^{n} T_i^2} * \sqrt{\sum_{k=1}^{n} S_i^2}}, \tag{1}$$

where T_i and S_i are the components of vector T and S, respectively. As a result, for each topic vector and sentence vector pair, we have an RSV.

3.5 Sentence ranking

The sentences are ranked in decreasing order of RSVs and delivered to the summary generation phase. The sentences which are most similar to the K topic vectors are included in the summary.

3.6 Summary generation

The final phase in our approach is summary generation. It entails removing redundancy from the highest-scoring sentences. The number of documents to be summarized in multi-document summarization can be

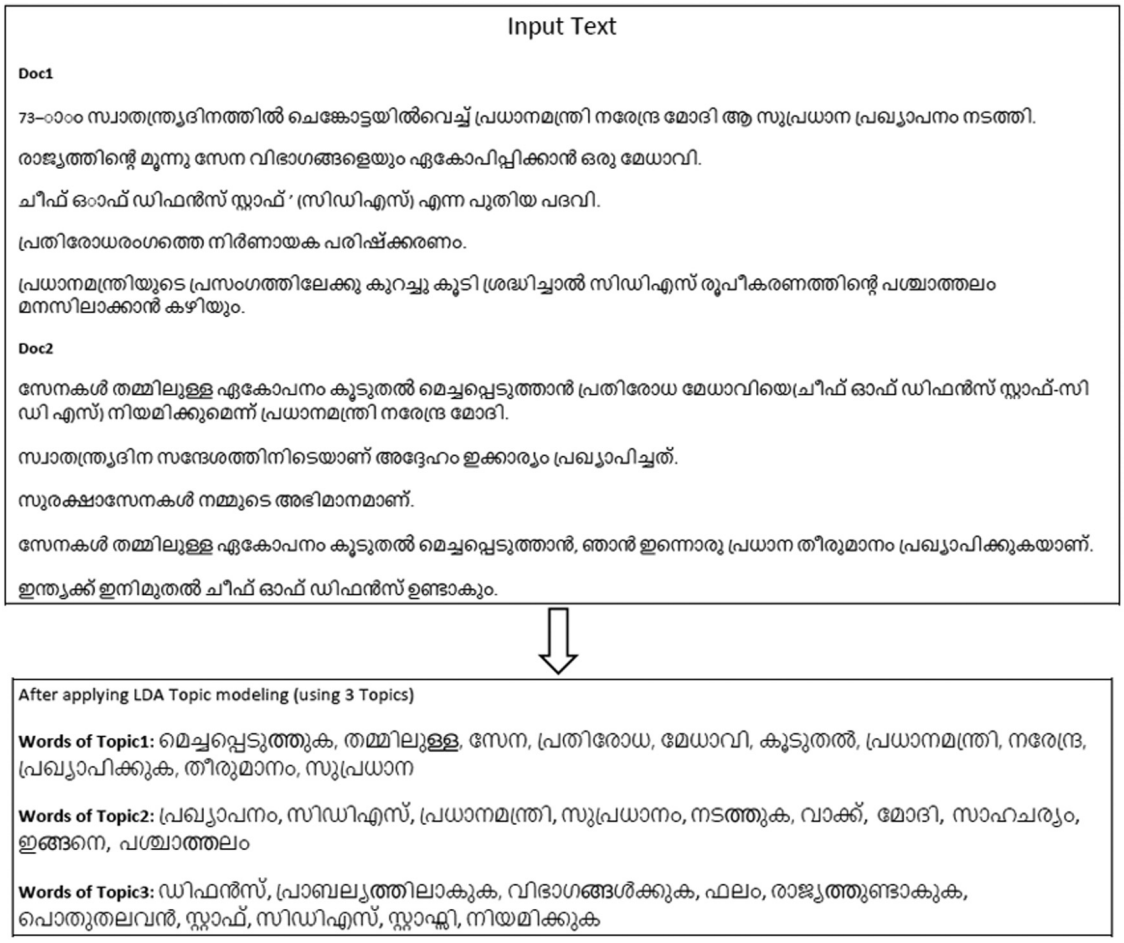

Figure 2: Example for Topic word generation from input text using LDA.

very high. As a result, the information redundancy in multi-document summarization is more severe than in single-document summarization. Controlling redundancy is essential. Maximal marginal relevance (MMR) [39] and clustering [40] are the two widely used methods for preventing duplication in summarization. The textual overlap between the sentence to be added to the output summary and the sentences in the generated summary text is used to determine redundancy in MMR. We start by inserting the first sentence from the ranking list into the summary. Then we look at the next one to see if it compares to the sentence(s) already in summary. Only the sentences that are not too similar to any of the sentences in summary (i.e., their cosine similarity is less than a threshold value of 0.66) are included in the summary. This process is repeated until the total length of the summary reaches the maximum length allowed. As a result, we can be confident that our final generated summary includes a wide range of information from the original input document.

Algorithm 1 provides the general phases of the proposed approach.

Algorithm 1: LDA with MMR-based MDS

Input

D: The merged document with n sentences, where $D = \{S_1, S_2, \ldots, S_n\}$

K: The no. of topics for LDA modeling. (User defined)

W: The no. of words to be included in each topic. (User defined)

C: The no. of sentences to be included in the summary. (User defined)

Output S': The extractive multidocument summary; where $S' \subset D$

1 Segment the document D into sentences and perform the preprocessing steps:
(i) Tokenization
(ii) Punctuation removal
(iii) Stopword removal
(iv) Stemming

2 Generate the vector for each sentence in the document D.

3 Generate topic vector of the document D using LDA modeling (top W words assigned to K topics)

4 Obtain the RSV for each sentence by determining the similarity between sentence vector and topic vector.

5 Rank the sentences according to RSV.

6 Perform MMR to generate the final summary S':
(i) Add the first sentence from the Ranklist to summary
(ii) Compare the next sentence with the existing sentences in the summary
(iii) If the similarity of the sentence to be added with existing summary sentences is less than 0.66 then the sentence is added to S'.
Repeat steps (ii)–(iii) till the *length* $(S') < C$.

4 Performance evaluation

4.1 Evaluation metric

We used the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [34] for evaluation, namely ROUGE-N (ROUGE-1 and ROUGE-2), ROUGE-L, and ROUGE-SU4. ROUGE is a totally automated and cutting-edge method for evaluating text summarization. ROUGE-N determines the degree of similarity between the system summary and the reference summary based on the n-gram comparison and overlap. It is calculated as follows:

$$\text{ROUGE-N} = \frac{\sum_{S\in(\text{Ref: Summary})}\sum_{\text{gram}_N\in(S)}\text{Count}_{\text{match}}(\text{gram}_N)}{\sum_{S\in(\text{Ref: Summary})}\sum_{\text{gram}_N\in(S)}\text{Count}(\text{gram}_N)}, \tag{2}$$

where N stands for the length of the N-gram and $\text{Count}_{\text{match}}(\text{gram}_N)$ is the maximum number of N-grams that occur in both reference summary and candidate summary. The most commonly used ROUGE measures are ROUGE-1 and ROUGE-2, which determine the number of overlapping unigrams and bigrams, respectively. ROUGE-SU4 is a variant of ROUGE-SU, which is an enhanced version of ROUGE-S (Skip bigram) [41]. ROUGE-SU4 skips a maximum distance of 4 between the bigrams used. ROUGE-L evaluates the fluency of the summary using the longest common subsequence (LCS) method, which considers sentence-level structure similarity. Let S be the system summary and R the reference summary that contains n words. ROUGE-L is calculated as follows:

$$\text{ROUGE-L} = \frac{\text{LCS}(S, R)}{n}, \tag{3}$$

where $\text{LCS}(S, R)$ is the length of the longest subsequence of S and R.

4.2 Experiments and results

All experimental processes were performed using a computer with an Intel Core i5-8250 CPU 1.80 GHz and 8 GB RAM using Python.

Experiments were conducted on the dataset designed explicitly for summarization to get a detailed evaluation of the proposed LDA-based MDS system. We ran the model with 10, 20, 30, and 40% compression ratios (CRs). Furthermore, the performance of the suggested model was examined by altering the number of topics to 3, 5, and 9. Table 2 shows the results of the proposed MDS in terms of Recall, Precision, and F-measure for 10% CR. According to the table, the more compressed data with fewer topics have an excellent ROUGE score value and it gradually decreases when more topics are covered. Table 3 shows the ROUGE-1 and ROUGE-2 scores for different CRs on varying the number of topics as 3, 5, and 9. From the table, it can be seen that the model with nine topics performs better than other models.

Table 2: ROUGE values for the proposed model on varying the number of topics with 10% CR

CR = 10%					
# Topics	**Measures**	**ROUGE-L**	**ROUGE-1**	**ROUGE-2**	**ROUGE-SU4**
3	Recall	0.33182	0.29502	0.28632	0.29163
	Precision	0.82022	0.81915	0.78824	0.79737
	F-score	*0.47249*	0.4338	0.42006	0.42706
5	Recall	0.29091	0.27586	0.23932	0.24543
	Precision	0.64	0.6729	0.59574	0.60714
	F-score	0.4	0.3913	0.34146	0.34955
9	Recall	0.28636	0.2567	0.23504	0.2435
	Precision	0.64948	0.64423	0.56122	0.56854
	F-score	0.39748	0.36712	0.33133	0.34097

Table 3: ROUGE-1 and ROUGE-2 values for the proposed model on varying the number of topics for different CRs

		CR 10%		**CR 20%**		**CR 30%**		**CR 40%**	
# Topics	**Measures**	**ROUGE 1**	**ROUGE 2**	**ROUGE 1**	**ROUGE 2**	**ROUGE 1**	**ROUGE 2**	**ROUGE 1**	**ROUGE 2**
3	Recall	0.29502	0.28632	0.46743	0.42735	0.68199	0.65385	0.75862	0.7265
	Precision	0.81915	0.78824	0.58095	0.52083	0.55799	0.52577	0.54848	0.51672
	F-score	0.4338	0.42006	0.51805	0.46948	0.61379	0.58286	0.63666	0.60391
5	Recall	0.27586	0.23932	0.57471	0.54701	0.62835	0.59402	0.72414	0.68803
	Precision	0.6729	0.59574	0.65502	0.61244	0.57143	0.53053	0.5431	0.50789
	F-score	0.3913	0.34146	0.61224	0.57788	0.59854	0.56048	0.62069	0.58439
9	Recall	0.2567	0.23504	0.47127	0.44444	0.69349	0.66667	0.7931	0.77778
	Precision	0.64423	0.56122	0.62121	0.55026	0.58766	0.55516	0.59312	0.57233
	F-score	0.36712	0.33133	0.53595	0.49173	0.6362	0.60583	*0.67869*	*0.65942*

The values in italics show the highest scores among the numbers in the table.

An experimental process was created in which the LDA-based MDS was compared with the Textrank-based MDS. Table 4 compares the ROUGE-1 values of the models for different CRs before eliminating the redundancy. This demonstrates that projecting sentences into topic space provides a more accurate representation of the input Malayalam documents while also delivering relevant information. When comparing the ROUGE-1 results of the 10% CR with the results of the 40% CR, from Table 4 it can be concluded that the F-score drops when the CR goes down because of the decrease in co-occurrence between the system summary and the reference summary.

The quality of text summarization is increased by eliminating redundancy. The accuracy of the final summary is improved considerably when a redundancy elimination component is used. The MMR algorithm was used to remove redundancy and improve information richness. A summary is obtained with less redundancy and more diversity in information on applying MMR to the rankings obtained after working with the LDA model. When comparing the findings in Tables 4 and 5, it is clear that removing redundancy enhances the quality of the final summary.

Table 4: ROUGE 1 values for the models for different CRs before redundancy removal

Model	CR 10%	CR 20%	CR 30%	CR 40%
Text Rank				
Recall_avg	0.2222	0.3295	0.40909	0.49573
Precision_avg	0.63736	0.53086	0.54878	0.42963
F-score_avg	0.32955	0.40662	*0.46875*	0.46032
LDA Model (Proposed system)# Topics: 9				
Recall_avg	0.2567	0.47127	0.69349	0.7931
Precision_avg	0.64423	0.62121	0.58766	0.59312
F-score_avg	0.36712	0.53595	0.6362	*0.67869*

The values in italics show the highest scores among the numbers in the table.

Table 5: ROUGE 1 values for the models for different CRs after redundancy removal

Model	CR 10%	CR 20%	CR 30%	CR 40%
Text Rank				
Recall_avg	0.27044	0.32143	0.50649	0.50649
Precision_avg	0.47253	0.61111	0.56727	0.51316
F-score_avg	0.344	0.42128	*0.53516*	0.5098
LDA Model (Proposed system)# Topics: 9				
Recall_avg	0.27586	0.48659	0.71648	0.75
Precision_avg	0.6729	0.61353	0.60129	0.62738
F-score_avg	0.3913	0.54274	0.65385	*0.68323*

The values in italics show the highest scores among the numbers in the table.

Figure 3 depicts Malayalam document summarization using topic model representation and MMR for summary creation. A document collection is randomly selected including two news items to demonstrate sentence ranking using LDA and the extractive summary of our algorithm. As can be seen, the suggested technique assigns the highest rank to Sentence 1 of Document 2, which is referred to as 2_1. This is because the LDA methodology generates "*pradhanamanthri*," "*mechapeduthuka*," "*pradhirodha*," "*medhavi*" as

DOCUMENT	Doc1	Doc2	Sentence Ranked after LDA modeling	Summary after redundancy removal with MMR
MALAYALAM	73–ാം സ്വാതന്ത്ര്യദിനത്തിൽ ചെങ്കോട്ടയിൽവെച്ച് പ്രധാനമന്ത്രി നരേന്ദ്ര മോദി ആ സുപ്രധാന പ്രഖ്യാപനം നടത്തി. [1] രാജ്യത്തിന്റെ മൂന്നു സേന വിഭാഗങ്ങളെയും ഏകോപിപ്പിക്കാൻ ഒരു മേധാവി. [2] ചീഫ് ഒാഫ് ഡിഫൻസ് സ്റ്റാഫ് ' (സിഡിഎസ്) എന്ന പുതിയ പദവി. [3] പ്രതിരോധരംഗത്തെ നിർണായക പരിഷ്കരണം.[4] പ്രധാനമന്ത്രിയുടെ പ്രസംഗത്തിലേക്കു കുറച്ചു കൂടി ശ്രദ്ധിച്ചാൽ സിഡിഎസ് രൂപീകരണത്തിന്റെ പശ്ചാത്തലം മനസിലാക്കാൻ കഴിയും.[5]	സേനകൾ തമ്മിലുള്ള ഏകോപനം കൂടുതൽ മെച്ചപ്പെടുത്താൻ പ്രതിരോധ മേധാവിയെ(ചീഫ് ഓഫ് ഡിഫൻസ് സ്റ്റാഫ്-സി ഡി എസ്) നിയമിക്കുമെന്ന് പ്രധാനമന്ത്രി നരേന്ദ്ര മോദി.[1] സ്വാതന്ത്ര്യദിന സന്ദേശത്തിനിടെയാണ് അദ്ദേഹം ഇക്കാര്യം പ്രഖ്യാപിച്ചത്. [2] സുരക്ഷാസേനകൾ നമ്മുടെ അഭിമാനമാണ്.[3] സേനകൾ തമ്മിലുള്ള ഏകോപനം കൂടുതൽ മെച്ചപ്പെടുത്താൻ, ഞാൻ ഇന്നൊരു പ്രധാന തീരുമാനം പ്രഖ്യാപിക്കുകയാണ്. [4] ഇന്ത്യക്ക് ഇനിമുതൽ ചീഫ് ഓഫ് ഡിഫൻസ് ഉണ്ടാകും.[5]	സേനകൾ തമ്മിലുള്ള ഏകോപനം കൂടുതൽ മെച്ചപ്പെടുത്താൻ പ്രതിരോധ മേധാവിയെ ചീഫ് ഓഫ് ഡിഫൻസ് സ്റ്റാഫ്-സി ഡി എസ്) നിയമിക്കുമെന്ന് പ്രധാനമന്ത്രി നരേന്ദ്ര മോദി.[2_1] സേനകൾ തമ്മിലുള്ള ഏകോപനം കൂടുതൽ മെച്ചപ്പെടുത്താൻ, ഞാൻ ഇന്നൊരു പ്രധാന തീരുമാനം പ്രഖ്യാപിക്കുകയാണ്.[2_4] 73–ാം സ്വാതന്ത്ര്യദിനത്തിൽ ചെങ്കോട്ടയിൽവെച്ച് പ്രധാനമന്ത്രി നരേന്ദ്ര മോദി ആ സുപ്രധാന പ്രഖ്യാപനം നടത്തി.[1_1] പ്രധാനമന്ത്രിയുടെ പ്രസംഗത്തിലേക്കു കുറച്ചു കൂടി ശ്രദ്ധിച്ചാൽ സിഡിഎസ് രൂപീകരണത്തിന്റെ പശ്ചാത്തലം മനസിലാക്കാൻ കഴിയും.[1_5] 'ചീഫ് ഒാഫ് ഡിഫൻസ് സ്റ്റാഫ് ' (സിഡിഎസ്) എന്ന പുതിയ പദവി.[1_3]	സേനകൾ തമ്മിലുള്ള ഏകോപനം കൂടുതൽ മെച്ചപ്പെടുത്താൻ പ്രതിരോധ മേധാവിയെ (ചീഫ് ഓഫ് ഡിഫൻസ് സ്റ്റാഫ്-സി ഡി എസ്) നിയമിക്കുമെന്ന് പ്രധാനമന്ത്രി നരേന്ദ്ര മോദി.[2_1] പ്രധാനമന്ത്രിയുടെ പ്രസംഗത്തിലേക്കു കുറച്ചു കൂടി ശ്രദ്ധിച്ചാൽ സിഡിഎസ് രൂപീകരണത്തിന്റെ പശ്ചാത്തലം മനസിലാക്കാൻ കഴിയും.[1_5] 'ചീഫ് ഒാഫ് ഡിഫൻസ് സ്റ്റാഫ് ' (സിഡിഎസ്) എന്ന പുതിയ പദവി.[1_3] 73–ാം സ്വാതന്ത്ര്യദിനത്തിൽ ചെങ്കോട്ടയിൽവെച്ച് പ്രധാനമന്ത്രി നരേന്ദ്ര മോദി ആ സുപ്രധാന പ്രഖ്യാപനം നടത്തി.[1_1]
ENGLISH (Google Translated)	Prime Minister Narendra Modi made that important announcement at the Red Fort on the 73rd Independence Day. [1] A commander to coordinate the three armies of the country. [2] New rank of Chief of Defence Staff (CDS). [3] Critical reforms in the defence sector.[4] A closer look at the Prime Minister's speech reveals the context behind the formation of the CDS. [5]	Prime Minister Narendra Modi will appoint the Chief of Defence Staff (CDS) to further improve coordination between the forces. [1] He made the announcement during his Independence Day message. [2] The security forces are our pride. [3] To further improve the coordination between the forces, I am announcing an important decision today. [4] India will now have a Chief of Defense. [5]	Prime Minister Narendra Modi will appoint the Chief of Defence Staff (CDS) to further improve coordination between the forces. [2_1] To further improve the coordination between the forces, I am announcing an important decision today. [2_4] Prime Minister Narendra Modi made that important announcement at the Red Fort on the 73rd Independence Day. [1_1] A closer look at the Prime Minister's speech reveals the context behind the formation of the CDS. [1_5] New rank of Chief of Defence Staff (CDS). [1_3]	Prime Minister Narendra Modi will appoint the Chief of Defence Staff (CDS) to further improve coordination between the forces. [2_1] A closer look at the Prime Minister's speech reveals the context behind the formation of the CDS. [1_5] New rank of Chief of Defence Staff (CDS). [1_3] Prime Minister Narendra Modi made that important announcement at the Red Fort on the 73rd Independence Day. [1_1]

Figure 3: Example for Summary generation by the proposed system.

topic words with high probability (as shown in Figure 2), and hence sentence 2_1 has the most similarity with the topic vector constructed using these terms. The proposed methodology incorporates the sentences that cover the majority of the topic words in the input document in the summary. After the proposed model calculates the first rank of each sentence, the MMR algorithm is used to re-rank the sentences and eliminate redundant information, which consists of deleting unwanted sentences that are identical to already picked sentences. Sentences 2_1 and 2_4 contain the same information. When MMR is used, the final summary is devoid of repetition.

The proposed technique was tested using the DUC-2003 dataset created by NIST[2] (National Institute of Standards and Technology), to assess the multi-document summarizing task for English. Table 6 displays the ROUGE-1 result for DUC2003. The results suggest that the proposed technique works well with the English language.

Table 6: ROUGE score comparison on DUC-2003 English dataset

Model	ROUGE-1	ROUGE-2	ROUGE-L
Textrank	0.44703	0.20462	0.21490
Proposed model	0.48821	0.22471	0.24968

To assess the performance of the proposed method, it was compared to some of the previous works on summarization in Indian languages. In ref. [42], the authors experimented on 100 news articles in Hindi, Punjabi, Marathi, and Tamil language using the four techniques used in Indian languages. They are Graph-based technique for Hindi, a hybrid model for Punjabi, Textrank-based technique for Marathi, and semantic graph-based method for Tamil. Table 7 demonstrates that our proposed method outperforms previous research in various Indian languages, despite the fact that the other languages' studies were done on single documents, but this offers a reference point. It is evident from the table that our model outperformed all existing studies in Indian languages. Our method for extractive summarization of multiple Malayalam documents has been demonstrated to be efficient.

Table 7: ROUGE-1 Precision, Recall, F-score comparison with previous research in Indian languages

Methods	Language	Precision	Recall	F-score
Graph based [43]	Hindi	0.44	0.32	0.37
Hybrid [19]	Punjabi	0.45	0.21	0.29
Textrank [21]	Marathi	0.43	0.27	0.33
Semantic graph [22]	Tamil	0.42	0.31	0.35
Proposed model	Malayalam	*0.63*	*0.75*	*0.68*

The values in italics show the highest scores among the numbers in the table.

5 Conclusion

This article proposes a topic modeling method for automatic extractive multi-document summarization for Malayalam documents. LDA-based topic models are employed to extract dominating topic terms from texts. The proposed method makes a significant addition by presenting a generalized mechanism that replaces the provided documents into a reduced size dimension with the topic vector that determines the relevance

2 https://duc.nist.gov

of the sentences. Furthermore, a redundancy removal component was used to diversify the contents in the final summary.

A dataset for MDS in Malayalam that consists of clusters of news articles paired with summaries of news events, was used without which it would be impossible to complete this study. Later on, experiments have been performed on two MDS datasets, Malayalam and English, individually to ensure the effectiveness of the automated summary of the proposed model. The experimental analysis revealed that the results of the strategy were more encouraging and stimulating than the baseline model. This article presented a comparative analysis with the summarization works done in Indian Languages to conclude that LDA gives a higher ROUGE value than other text summarization techniques used in Indian Languages.

Although the proposed model automatically generated a non-redundant summary from several sources, it was not as coherent as a human summary. However, most of the time, the readers could comprehend the summary. Even though the work has been implemented and evaluated for the Malayalam language, it can also be adapted for any other language. In future, it is proposed to incorporate topic modeling methodology in other techniques of extractive text summarization like graph-based method and evolutionary algorithms.

Acknowledgements: The authors thank Dr. Saidalavi C. and Ms. Prajisha A. K. from the Department of linguistics, Thunchath Ezhuthachan Malayalam University, for validating the summary during the dataset preparation process.

Conflict of interest: Authors state no conflict of interest.

References

[1] Widyassari AP, Rustad S, Shidik GF, Noersasongko E, Syukur A, Affandy A, et al. Review of automatic text summarization techniques and methods. J King Saud Univ Comput Inform Sci. 2020. doi: 10.1016/j.jksuci.2020.05.006.
[2] Radev DR, Jing H, Styś M, Tam D. Centroid-based summarization of multiple documents. Inform Process Manag. 2004;40(6):919–38. ISSN 0306-4573. doi: 10.1016/j.ipm.2003.10.006.
[3] Mao X, Yang H, Huang S, Liu Y, Li R. Extractive summarization using supervised and unsupervised learning. Expert Syst Appl. 2019;133:173–81, ISSN 0957-4174. doi: 10.1016/j.eswa.2019.05.011.
[4] Yau CK, Porter A, Newman N, Suominen A. Clustering scientific documents with topic modeling. Scientometrics 2014 Sep 1;100(3):767–86.
[5] Jelodar H, Wang Y, Yuan C, Feng X, Jiang X, Li Y, et al. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. Multimedia Tools Appl. 2019 Jun;78(11):15169–211.
[6] Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. J Machine Learn Res. 2003;3:993–1022.
[7] Arora R, Ravindran B. Latent Dirichlet allocation based multi-document summarization. In: Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data; 2008 Jul 24. p. 91–7.
[8] Twinandilla S, Adhy S, Surarso B, Kusumaningrum R. Multi-document summarization using k-means and latent Dirichlet allocation (LDA)-significance sentences. Proc Comput Sci. 2018;135:663–70.
[9] Yang G, Wen D, Chen NS, Sutinen E. A novel contextual topic model for multi-document summarization. Expert Syst Appl. 2015 Feb 15;42(3):1340–52.
[10] Rani R, Lobiyal DK. An extractive text summarization approach using tagged-LDA based topic modeling. Multimedia Tools Appl. 2021 Jan;80(3):3275–305.
[11] Rani U, Bidhan K. Comparative assessment of extractive summarization: textrank TF-IDF and LDA. J Sci Res. 2021;65(1):304–11.
[12] Radev DR, Allison T, Blair-Goldensohn S, Blitzer J, Celebi A, Dimitrov S, et al. MEAD-a platform for multidocument multilingual text summarization. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04). Lisbon, Portugal: European Language Resources Association (ELRA); 2004.
[13] Al-Radaideh QA, Bataineh DQ. A hybrid approach for Arabic text summarization using domain knowledge and genetic algorithms. Cognitive Comput. 2018 Aug;10(4):651–69.
[14] Elbarougy R, Behery G, ElKhatib A. Extractive Arabic text summarization using modified PageRank algorithm. Egypt. Inform J 2020 Jul 1;21(2):73–81.
[15] Xi X, Pi Z, Zhou G. Global encoding for long Chinese text summarization. ACM Trans Asian Low-Resource Language Inform Process (TALLIP). 2020 Oct 6;19(6):1–7.

[16] Kumar Y, Kaur K, Kaur S. Study of automatic text summarization approaches in different languages. Artif Intell Rev. 2021 Feb 12;54(8):1–33.
[17] Bouscarrat L, Bonnefoy A, Peel T, Pereira C. STRASS: A light and effective method for extractive summarization based on sentence embeddings. 2019 Jul 16. arXiv preprint: http://arXiv.org/abs/arXiv:1907.07323.
[18] Gambhir M, Gupta V. Recent automatic text summarization techniques: a survey, Artif Intell Rev. 2017;47:1–66. doi: 10.1007/s10462-016-9475-9.
[19] Gupta V, Kaur N. A novel hybrid text summarization system for Punjabi text. Cognitive Comput. 2016 Apr 1;8(2):261–77.
[20] Nawaz A, Bakhtyar M, Baber J, Ullah I, Noor W, Basit A. Extractive text summarization models for Urdu language. Inform Process Manag. 2020 Nov 1 57(6):102383.
[21] Rathod TV. Extractive text summarization of Marathi news articles. IRJET. 2018;5:1204–10.
[22] Banu M, Karthika C, Sudarmani P, Geetha TV. Tamil document summarization using semantic graph method. In: International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007). vol. 2. IEEE; 2007 Dec 13. p. 128–34.
[23] Manju K, DavidPeter S, Idicula SM. A framework for generating extractive summary from multiple Malayalam documents. Information. 2021 Jan;12(1):41.
[24] Hovy E, Lin CY. Automated text summarization in SUMMARIST. Adv Automatic Text Summarization. 1999;14:81–94.
[25] Zhang J, Tan J, Wan X. Adapting neural single-document summarization model for abstractive multi-document summarization: a pilot study. In: Proceedings of the 11th International Conference on Natural Language Generation; 2018 Nov. p. 381–90.
[26] Belwal RC, Rai S, Gupta A. Text summarization using topic-based vector space model and semantic measure. Inform Process Manag. 2021 May 1;58(3):102536.
[27] Kumar R, Raghuveer K. Legal document summarization using latent Dirichlet allocation. Int J Comput Sci Telecommun. 2012;3(114–117):8–23.
[28] Erkan G, Radev D. Lexpagerank: Prestige in multi-document text summarization. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing; 2004 Jul. p. 365–71.
[29] Mihalcea R, Tarau P. Textrank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing; 2004 Jul. p. 404–11.
[30] Barrera A, Verma R. Combining syntax and semantics for automatic extractive single-document summarization. In: International Conference on Intelligent Text Processing and Computational Linguistics. Berlin, Heidelberg: Springer; 2012 Mar 11. p. 366–77.
[31] Uçkan T, Karcı A. Extractive multi-document text summarization based on graph independent sets. Egypt Inform J. 2020 Sep 1;21(3):145–57.
[32] Nallapati R, Zhai F, Zhou B. Summarunner: a recurrent neural network based sequence model for extractive summarization of documents. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. February 4–9, 2017, San Francisco, California, USA. p. 3075–81.
[33] Al-Sabahi K, Zuping Z, Nadher M. A hierarchical structured self-attentive model for extractive document summarization (HSSAS). IEEE Access. 2018 Apr 23;6:24205–12.
[34] Lin CY. Rouge: a package for automatic evaluation of summaries. In: Text summarization branches out; 2004, Jul. p. 74–81.
[35] Tran NT, Nghiem MQ, Nguyen NT, Nguyen NLT, Van Chi N, Dinh D. ViMs: a high-quality Vietnamese dataset for abstractive multi-document summarization. Language Resour Evaluat. 2020;54(4):893–920.
[36] Radev D, Teufel S, Saggion H, Lam W, Blitzer J, Qi H, et al. Evaluation challenges in large-scale document summarization. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. 2003 Jul. p. 375–82.
[37] Thottungal S. Indic Stemmer. 2019. Available online: https://silpa.readthedocs.io/projects/indicstemmer (accessed on March 12, 2019).
[38] Gialitsis N, Pittaras N, Stamatopoulos P. A topic-based sentence representation for extractive text summarization. In: Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources; 2019 Sep. p. 26–34.
[39] Goldstein J, Carbonell JG. Summarization: (1) using MMR for diversity-based reranking and (2) evaluating summaries. In TIPSTER TEXT PROGRAM PHASE III: Proceedings of a Workshop held at Baltimore, Maryland; October 13–15, 1998. p. 181–195.
[40] Radev DR, Jing H, Styś M, Tam D. Centroid-based summarization of multiple documents. Inform Process Manag. 2004 Nov 1;40(6):919–38.
[41] Saziyabegum S, Sajja PS. Review on text summarization evaluation methods. Indian J Comput Sci Eng. 2017;8(4):497500.
[42] Verma P, Verma A. Accountability of NLP tools in text summarization for Indian languages. J Scient Res. 2020;64(1):358–63.
[43] Kumar KV, Yadav D. An improvised extractive approach to Hindi text summarization. In: Information Systems Design and Intelligent Applications. New Delhi: Springer; 2015. p. 291–300.