Journal of Information Technology Research
Volume 15 • Issue 1

Automatic Text Summarization by Providing Coverage, Non-Redundancy, and Novelty Using Sentence Graph

Krishnaveni P., National Institute of Technology, Trichy, India
Balasundaram S. R., National Institute of Technology, Trichy, India

ABSTRACT

The day-to-day growth of online information necessitates intensive research in automatic text summarization (ATS). The ATS software produces summary text by extracting important information from the original text. With the help of summaries, users can easily read and understand the documents of interest. Most of the approaches for ATS used only local properties of text. Moreover, the numerous properties make the sentence selection difficult and complicated. So this article uses a graph-based summarization to utilize structural and global properties of text. It introduces maximal clique-based sentence selection (MCBSS) algorithm to select important and non-redundant sentences that cover all concepts of the input text for summary. The MCBSS algorithm finds novel information using maximal cliques (MCs). The experimental results of recall-oriented understudy for gisting evaluation (ROUGE) on Timeline dataset show that the proposed work outperforms the existing graph algorithms: bushy path (BP), aggregate similarity (AS), and textrank (TR).

KEYWORDS

Graph-Based Summarization, Isolated Sentences, Local Properties of Text, Maximal Clique-Based Sentence Selection, Novel Information, ROUGE, Structural and Global Properties of Text, Timeline Dataset

INTRODUCTION

Due to the huge availability of online information, comprehending and assimilating the vast information is a major problem. The ATS process can be used to obtain a detailed insight of either a single document or a group of documents. This article aims to produce a single document summary. The summary is a short text that is produced from the original text. It should contain important information and no redundant information. ATS is a two class classification problem of machine learning. It classifies the entire text into summary sentences and non-summary sentences. The work of Luhn (1958) introduced ATS of a single document. Later it evolved gradually by various approaches.

The two major types of summarization process are extractive summarization and abstractive summarization. Since abstractive summarization requires artificial intelligence, extractive summarization can be used widely. In single document summarization, sentence scoring techniques are the most popular at the beginning. The three kinds of sentence scoring techniques are word scoring,

DOI: 10.4018/JITR.2022010108

sentence scoring, and graph scoring (Ferreira et al., 2013a). The word scoring and sentence scoring techniques depend only on local word specific and sentence specific information respectively. But they fail to utilize global, group, and structural information. To resolve this, the authors go for graph based summarization.

In a textual graph, each node (sentence) represents one text unit and an edge (link) between two nodes represents the relationship between them. A text unit may be a word, sentence, phrase, or paragraph. In a sentence graph, there is a link between two sentences only if they have some common information. So high frequency words, excluding stop words would link a number of sentences.

The authors produce a summary of the given input document using the sentence graph and its maximal cliques (MCs). A *clique* is a complete sub graph of a graph. All nodes in a clique related to each other and share similar information. The *maximal clique* (MC) of a graph is a clique which is not a proper subset of any other clique (Tomita, Akutsu, & Matsunaga, 2011). The clique in a graph could be considered as a representation of a cluster, but the sentences are not restricted to belong to exactly one clique. This is the major difference between a clique and a cluster (Nenkova & McKeown, 2011). Thus, this article combines the benefits of both word frequency and sentence clustering methods. It takes the advantageous of overlapping of cliques to select summary sentences.

The aim of this article is to produce a generic graph based extractive single document summary with minimum number of informative sentences. The structure of this article is framed as follows. First, this article discusses the background of ATS. Then, it describes the proposed work, its implementation, and explanation with one sample text. Next, it explains the evaluation methods, experimental design, and experimental results. Next, the authors conclude the proposed work and suggest the work that can be carried out in the future. Finally, this article lists the references cited in this work.

BACKGROUND

In a graph based summarization model, a node can be scored using information from the global graph. First, Mani and Bloedorn (1997) proposed graph representation of text. They described a new mechanism for summarizing the similarities and differences between a pair of related documents. Another work used the knowledge of text structure for producing summaries by automatic passage extraction (Salton, Singhal, Mitra, & Buckley, 1997). The earlier iterative graph algorithms are TextRank (Mihalcea & Tarau, 2004) and LexRank (Erkan & Radev, 2004). They can be applied to the summarization of a single or multiple documents in any language (Mihalcea & Tarau, 2005). Even though they are the best graph ranking algorithms, they have high time complexity. Some recent graph based ranking research works are Calvo et al. (2018), Feiyue and Xinchen (2018), and Tixier et al. (2017).

The work of Sornil and Gree-ut (2006) constructs an undirected document graph from cosine similarity using the Hopfield Network algorithm for text segment ranking. One research work produces automatic summaries using graph algorithms and the shortest path algorithm and compares them (Khushboo, Dharaskar, & Chandak, 2010). It concludes that the shortest path algorithm is the best one, because it generates a smooth summary. The work of Chen and Zhao (2014) constructs a two layer, phrase-sentence graph. It utilizes sentence relevance and phrase relevance information to produce a summary. The work of Zheng and Bai (2014) considers text summarization as a problem of finding the key paths composed of essential information and merges the common paths to remove redundancy. Han et al. (2016) propose the FrameNet-based semantic graph model. It uses FrameNet to calculate sentence similarity and assigns weights to both sentences and edges. After ranking, it selects summary sentences.

Oliveira et al. (2016) present a comparative analysis of eighteen sentence scoring features. They compute the importance of a sentence in extractive single and multi-document summarizations. The work of Verma and Om (2019) extracts the summary sentences with the help of a meta heuristic approach known as teaching-earning-based optimization. The work of Sariki et al. (2019) contains three different concurrent pipelines to improve the effectiveness of the summarization process. Simon et al. (2018) identify the best combinations of sentences that resemble human summary. Aries et al. (2018) use content based and graph based features to produce summary. Yang et al. (2018) use an integrated graph model to find the implicit semantic properties at the word level, bigram level, and trigram level. For every document, three different types of semantic graphs are obtained. They combine the three graphs into one enriched semantic graph. Finally, they rank the sentences using TR. The work of Ferreira et al. (2013b) proved experimentally that semantic similarity did not yield good results and cosine similarity achieved good results in ATS. So this article uses cosine similarity to construct the sentence graph.

Recently, algorithmic approaches are used to solve the ATS problem. One approach considers ATS as a maximum coverage problem with Knapsack constraints (Takamura & Okumura, 2009). It models the summary to cover each concept of the text. Another approach generates multi-document summary from a set of documents using the idea of graph domination (Shen & Li, 2010). The work of John and Wilscy (2015) models ATS as a vertex cover problem which produces a set of summary sentences that cover the entire concept of the text. These algorithmic approaches produce a summary that covers the predominant concepts of the input document, but the sentences in a concept are not necessarily connected to each other. The MCBSS approach selects summary sentences from a group of interrelated sentences, so it produces highly informative summary.

PROPOSED WORK

Overview

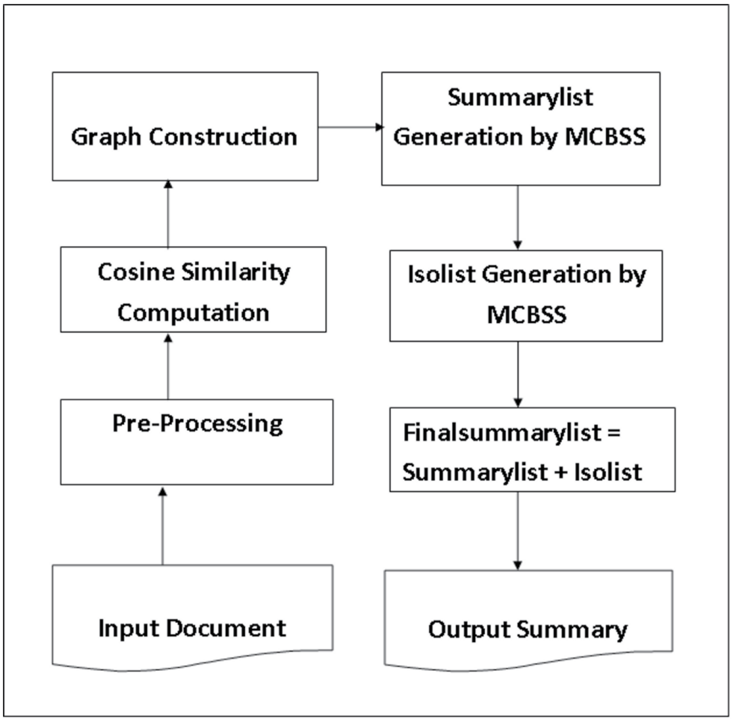
The proposed work designs the MCBSS algorithm to select the summary sentences that cover all concepts of the input text without redundancy and with novelty using the sentence graph. *It is considered as a problem of finding the minimum number of sentences that cover all MCs of the sentence graph*. It uses the heuristic that a sentence links with more sentences and/or more groups contains more information. This work uses the heuristic that a sentence links with more sentences and/or more groups contains more information. Since all sentences in an MC are related to each other, either a single independent MC or a group of related MCs represent one concept or main idea of the input text. This work takes one summary sentence from each concept of the text. The importance of a sentence in each MC can be identified by considering the number of cliques (groups) in which the sentence occurs. The sentence in more cliques is chosen as the summary sentence. Each summary sentence is a representative of information for all the MCs in which it occurs. Thus, the summary sentences convey the information contained on all the concepts in the input document. The union of the summary sentences with their group members gives the entire input document. This work avoids redundancy by taking summary sentences from distinct MCs.

This work has three modules which are Preprocessing, Graph construction, and Summary generation. The Preprocessing step transforms the input text into an intermediate form suitable for generating summary. The Graph construction step constructs a text relationship graph from the intermediate form of the input text. The Summary generation step generates summary text from the sentence graph using the MCBSS algorithm. Figure 1 shows the overall system architecture of the proposed work.

Preprocessing

Basic preprocessing operations are data cleaning, data integration, data transformation, and data reduction (Han & Kamber, 2006). They remove incomplete, inconsistent, and noisy data from the text document. In this work, the authors performed the preprocessing operations **C**ase Folding, Sentence Segmentation, Word Tokenization, Removal of Stop Words, and Word Stemming. These operations are used to improve the efficiency and accuracy of the summarization system. The authors used

Figure 1. System architecture



the Python package, Natural Language Tool Kit (NLTK) version 3.3 to preprocess the given input document. The NLTK is available at https://pypi.org.project/nltk/.

Graph Construction

This work computes a sentence similarity matrix using Cosine similarity and constructs a sentence graph using the Python package, NetworkX version 2.1 of the given preprocessed input text using this similarity matrix. NetworkX is available at https://pypi.org.project/networkx/. This graph is used to visualize the logical structure of the input document. Since Cosine similarity is a symmetric relation, the graph is undirected. An edge between two sentences is created in the graph, only if their Cosine similarity score is greater than or equal to a particular threshold value. Here, the authors used the threshold value of 0.1. Any link below the threshold value can be caused by random word matches between sentences and should not be considered as a valid link. Cosine similarity is used to measure the lexical similarity between two sentences. The formula for calculating Cosine similarity between any two sentences *S1* and *S2* is shown in equation (1):

$$Similarity\left(S1,S2\right)=\frac{No.\ of\ \ word\ \ overlap\ \ between\ \ S1\ \ and\ \ S2}{\sqrt{length\left(S1\right)}*\sqrt{length\left(S2\right)}} \quad (1)$$

Summary Generation

The MCBSS algorithm captures group relations among the sentences in the graph constructed. First, it gets a global list of all MCs in the graph. It uses the NetworkX function, find_cliques(G) to compute all maximal cliques of the sentence graph G in linear time. This function uses the Bron-Kerbosch algorithm to find all maximal cliques (School of Computing Science, 2013). A graph has at most $3^{n/3}$ maximal cliques, where n is the number of nodes of the graph *G*. So the time complexity of the Bron-Kerbosch algorithm is $O(3^{n/3})$. The time to compute all maximal cliques increases as *n* increases. The authors assume that the first sentence of the input document is important as said in the work of Fattah and Ren (2009). So, the MCBSS algorithm adds the first sentence of the input document in the *summarylist*. It finds the maximum clique size of the sentence graph using the NetworkX function, nx.graph_clique_number(G).This function returns the maximum clique size of the sentence graph *G*.

Second, it gets a local list of all cliques of maximum (largest) size using find_cliques(*G*). A *maximum clique* is a clique of maximum size, in the sense that no other clique contains more vertices. Third, the MCBSS algorithm selects the summary sentences from the largest size cliques, next largest size cliques, and so on until there are no more cliques in the global list or the clique size becomes one and stores them in *summarylist.* Later the MCBSS algorithm collects all the MCs of size one, which are called as isolated sentences of the graph and stores them in *isolist*. These sentences do not have any relationship with other sentences. So they represent novel information. Generally, they are very short sentences, only thirty percent of them can be selected as summary sentences based on the sentence length score. Because lengthy sentences are more informative than short sentences (Fattah & Ren, 2009). The sentence length score (SLS) of a sentence *S* is calculated using equation (2). The MCBSS algorithm for finding summary text from the textual graph is given in Algorithm 1.

$$SLS\left(S\right) = \frac{No.\ of\ words\ in\ S}{No. of words in longest sentence} \tag{2}$$

Algorithm 1: MCBSS

```
Input: The text relationship graph of the input document
Output: summarylist, isolist
1  List all maximal cliques of size greater than or equal to two
   in the text relationship graph and assign them to cliqs.  It is
   a global list.
2  Add node 0 (first sentence) to the summarylist, which is used
   to store the summary sentences. The cliques containing node 0
   are removed from cliqs. Other nodes except node 0 in these
   Cliques are added to the deletedlist, which is used to store
   the group members (adjacent nodes)of the summary sentences and
   avoids the selection of them again.
3  Find the maximum clique size of the graph and assign it to currentsize.
4  List all cliques of currentsize and assign them to currentcliqs.
   It is a local list.
5  Take the first clique from currentcliqs.
6  If the clique does not exist in cliqs, then it is a clique of
   already selected sentence. So it is removed from current cliqs.
   Go to step 9.
7  If all nodes of the clique are in deletedlist, simply remove
   the clique from cliqs and currentcliqs and go to step 9
   else for each node (sentence) in this clique which is not in
   deletedlist, find the number of cliques containing that node.
8  The node in more cliques is added to the summarylist. The
   cliques ontaining that node are removed from both cliqs and
```

```
   currentcliqs. Other nodes except the selected node in these
   cliques are added to the deletedlist, if it does not exist already.
9  If length of currentcliqs is greater than zero, go to step 5.
10 Decrease currentsize by one.
11 If currentsize is equal to one, go to step 13.
12 If length of cliqs is greater than zero, go to step 4.
13 List all cliques of size one and assign them to isolist.
14 Compute the sentence length score for each sentence in isolist.
15 Rearrange the isolist by decreasing order of sentence length score.
16 Now the summarylist contains non-redundant sentences and the
   isolist contains novel information.
```

The main algorithm combines the non-redundant and the isolated sentences to produce the final summary and sorts the final summary in the order of the original text. Now the summary contains the preprocessed sentences. It replaces the preprocessed sentences by their corresponding original sentences of the input document and displays the summary sentences. This work produces a thirty percent summary of the input text. The summary generated by this work is called the MCBSS summary. The authors also generate three more types of summaries using popular graph algorithms BP, AS, and TR for comparison purpose. The procedure for the entire summary generation process is given in Algorithm 2.

Algorithm 2: Main

```
1  Read the input document.
2  Preprocess the given input document.
3  Compute cosine similarity for every pair of sentences and
   construct a sentence similarity matrix.
4  Represent the preprocessed input text as a sentence graph using
   the similarity matrix.
5  Call MCBSS algorithm to select summary sentences without
   redundancy (summarylist) and to select summary sentences with
   novel(isolist) information.
6  Combine the two kinds of sentences produced by the MCBSS
   algorithm to produce the final summary (finalsummarylist).
7  Sort the final summary and replace the summary sentences with
   corresponding original sentences of the input document.
8  Display the summary sentences.
```

Analysis of the Proposed Work

The authors analyze the proposed work with the sample text, Document 6 (shortest) of the dataset used in this article. The sentence graph of the sample text is shown in Figure 2. The summary generation process is explained step by step in Figure 3. The sample text and its MCBSS summary are shown in Figure 4. Refer to Figure 3; the *finalsummarylist* contains the sentences 0, 3, 5, and 8. So the non-summary sentences of the input text are 1, 2, 4, 6, 7, 9, 10, 11, 12, and 13. Refer to Figure 2; the sentences 1, 2, 6, 7, 9, and 10 are adjacent to 5; 4 and 12 are adjacent to 0; 11and 13 are isolated sentences. Thus, *except isolated sentences every sentence of the input text is either in the summary or adjacent to a sentence in the summary*. So the summary sentences are enough to convey (cover) the overall information of the input text.

Refer to Figure 2; it is clearly understand that there is no direct link among the summary sentences. Also all cliques of the input text are: [[0, 2, 4, 7], [0, 12], [5, 6], [5, 7, 2, 1, 9], [5, 7, 2, 4], [5, 7, 10],

Figure 2. Sentence graph of sample text

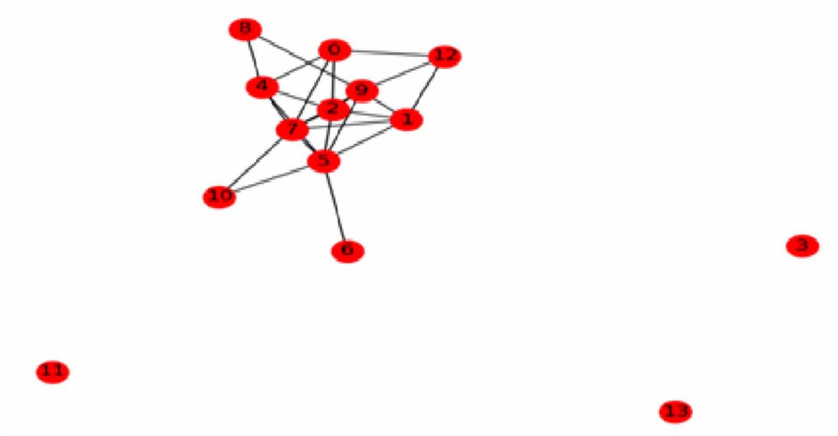


Figure 3. The summary generation process using MCBSS

<table>
<tr><th colspan="2">Non-redundant sentence selection</th></tr>
<tr><td colspan="2">Initially, all cliques of size >=2 are:
cliqs= [[0, 2, 4, 7], [0, 12], [5, 6], [5, 7, 2, 1, 9], [5, 7, 2, 4], [5, 7, 10], [8, 9], [8, 4], [12, 1, 9]]-global list
Node 0 is added to summarylist. Cliques containing node 0 are [[0, 2, 4, 7], [0, 12]] and are deleted from cliqs. Except node 0, the nodes 2, 4, 7, and 12 of these cliques are added to deletedlist.
Now summarylist : [0] deletedlist : [2, 4, 7, 12] Max_clique_size = 5</td></tr>
<tr><td>step: 0
cliqs =[[5, 6], [5, 7, 2, 1, 9], [5, 7, 2, 4], [5, 7, 10], [8, 9], [8, 4], [12, 1, 9]] length 7
currentsize =5 (Max_clique_size)
currentcliqs = Cliques of size 5 = [5, 7, 2, 1, 9] length=1
Clique taken: [5, 7, 2, 1, 9]. Now choose a node for summary that occurs in more cliques. maximum 4
Node 5 occurs in 4 cliques, so it is added to summarylist.
Cliques containing node 5: [[5, 6], [5, 7, 2, 1, 9], [5, 7, 2, 4], [5, 7, 10]] are deleted from both cliqs and currentcliqs. Except node 5, other nodes in these cliques are added to deleted list.
currentcliqs = [ ] length = 0
summarylist : [0, 5] deletedlist: [2, 4, 7, 12, 1, 9, 6, 10]</td>
<td>step: 1
cliqs = [[8, 9], [8, 4], [12, 1, 9]] length 3
currentsize =4
currentcliqs of size 4 are:[[0, 2, 4, 7], [5, 7, 2, 4]] length= 2
Clique taken: [0, 2, 4, 7]
It is not in cliqs (because it is a clique of already selected sentence). Hence, it is removed
currentcliqs = [5, 7, 2, 4] length = 1
Clique taken: [5, 7, 2, 4]
It is not in cliqs, so it is removed
currentcliqs = [ ] length= 0
summarylist: [0, 5]
deletedlist:[2, 4, 7, 12, 1, 9, 6, 10]</td></tr>
<tr><td>step: 2
cliqs [[8, 9], [8, 4], [12, 1, 9]] length 3
currentsize =3
currentcliqs of size3 are: [[5, 7, 10], [12, 1, 9]] length= 2
Clique taken: [5, 7, 10]
It is not in cliqs, so it is removed

currentcliqs = [12, 1, 9] length 1
Clique taken: [12, 1, 9]. All nodes are in deletedlist, that is the nodes12, 1, and 9 are the members of already selected sentences.
So [12, 1, 9] is removed from currentcliqs and cliqs.
currentcliqs= [ ] length= 0</td>
<td>step: 3
cliqs= [[8, 9], [8, 4]] length= 2
currentsize =2
currentcliqs of size 2 =[[0, 12], [5, 6], [8, 9], [8, 4]]length= 4
Clique taken: [0, 12]. It is not in cliqs, so it is removed
currentcliqs =[[5, 6], [8, 9], [8, 4]] length =3
Cliquetaken: [5, 6]. it is not in cliqs, so it is removed
currentcliqs =[[8, 9], [8, 4]] length= 2
Clique taken: [8, 9] maximum 2
Node 8 occurs in 2 cliques and is added to summarylist Cliques containing node 8: [[8, 9], [8, 4]] are deleted from both cliques. Except node 8, other nodes in these cliques are added to deleted list
summarylist: [0, 5, 8] deletedlist : [2, 4, 7, 12, 1, 9, 6, 10]
currentsize =1 exit</td></tr>
<tr><th colspan="2">Diverse or novel sentence selection</th></tr>
<tr><td colspan="2">Isolated sentences (cliques of size one) are: isolist: [3, 11, 13] length 3
isodict {3: 0.303, 11: 0.273, 13: 0.182}-dictionary contains sentence number-sentence length score pair [(3, 0.303), (11, 0.273), (13, 0.182)]- sorted by score
Take 30% of isolated sentences ((Having highest score) and add to summary. Here sentence 3 is taken as summary sentence. isolist=[3]
finalsummarylist = summarylist + isolist = [0, 5,8] + [3] =[0, 5, 8, 3]
Summary sentences are: [0, 5, 8, 3]. Summary sentences in sorted order: [0, 3, 5, 8]</td></tr>
</table>

Figure 4. Sample text and its MCBSS summary

Input Text

0. BP says oil spill cost up to $ 8bn The cost of the Gulf of Mexico oil disaster continues to mount up for BP BP says the cost of its Gulf of Mexico oil spill has risen to $ 8bn -LRB- # 5.2 bn -RRB- - a rise of more than $ 2bn in the last month alone .
1. The company said it had paid out about $ 399m in claims to those affected by the spill .
2. Last week , responsibility for the claims was transfered to the Gulf Coast Claims Facility -LRB- GCCF -RRB- , which has so far paid out a total of $ 38.5 m. Plans to permanently seal the well were also progressing well , BP said .
3. Are the safety practices in oil on a par with standard practice in nuclear generation or the airline industry ?
4. I would be very surprised if that reassuring conclusion will be drawn from -LRB- BP 's -RRB- Mr Bly 's report '' The final sealing of the well is now expected to be completed later this month .
5. BP said the capping stack placed on top of the well in July was removed on Thursday .
6. This will allow the failed blow-out preventer from the Deep Water Horizon rig to be removed , and a new blow-out preventer put in place .
7. BP said this would allow the drilling of the relief well to continue .
8. It is now expected to be completed in mid-September .
9. Meanwhile , the company said no new oil had flowed from the damaged well into the Gulf since the leak was stopped in mid-July .
10. BP shares rose slightly during the morning trading session .
11. They have risen by about 20 % since they hit a low of 303 pence in late June .
12. Separately , it was reported that the company is concerned that it may not be able to afford to pay for all the costs of the oil spill if US legislation bars it from applying for new offshore drilling permits .
13. The pun - ancient artform , or the lowest form of wit ?

MCBSS Summary

0. bp says oil spill cost up to $ 8bn the cost of the gulf of mexico oil disaster continues to mount up for bp bp says the cost of its gulf of mexico oil spill has risen to $ 8bn -lrb- # 5.2 bn -rrb- - a rise of more than $ 2bn in the last month alone .
1. are the safety practices in oil on a par with standard practice in nuclear generation or the airline industry ?
2. bp said the capping stack placed on top of the well in july was removed on thursday .
3. it is now expected to be completed in mid-september .

[8, 9], [8, 4], [12, 1, 9], [3], [11], [13]]. Since the summary sentences 0, 3, 5, and 8 are in distinct cliques, they share no common information and are disjointed. So there is no chance for redundancy. Also refer to Figure 4, the output summary contains no redundant information.

Refer to Figure 2 and Figure 3; the isolated sentences are 3, 11, and 13. The isolated sentence 3 is taken as the summary sentence because it has more sentence length score. So the summary provides novel or diverse information. Thus, the summary supports coverage and provides non-redundant and diverse information.

SUMMARY EVALUATION AND DISCUSSION

Overview

The generated summary should give an overview of the content of the entire document. So it should be evaluated. Summary evaluation is a challenging task, because there is no correct referencing answer due to human variation. There are two kinds of summary evaluation: intrinsic evaluation and extrinsic evaluation (Steinberger & Jezek, 2009). Intrinsic evaluation is mainly focused on the quality and the informativeness of the summary. Extrinsic evaluation is a task based evaluation. It tests the impact of summarization on tasks such as relevance assessment, reading comprehension, etc.

The authors have taken 10 input documents and their corresponding Timelines (human summaries) from the Timeline 17 dataset. The dataset is available at http://www.13s.de/~gtran/timeline/. It consists of 17 manually created Timelines and their associated news articles. It has 9 news topics. The authors have taken the topic BP Oil Spill. The Timelines and the news articles of this topic are available at the news agencies BBC and Guardian. Table 1 shows the various terms used in the experiment and their descriptions. Table 2 shows the statistics of the dataset used in the experiment.

The authors compared the proposed work with existing graph algorithms BP, AS, and TR for evaluating the summary. They have implemented the proposed work, BP, AS, and TR using the Python version 3.6.5. Python is available at https://www.python.org/downloads/. Also, they have taken textrank 0.1.0 from https://pypi.org.project/textrank/. It produces a hundred word summary of any given input document.

Baseline Systems

BP is a degree based approach. It selects the highest degree sentences as summary sentences. AS computes the score of a sentence by summing up the weights of the edges it has with other sentences. The highest score sentences are selected for summary. TR works by taking the global information recursively calculated from the entire textual graph. Text rank score of a node is determined based on the votes for it, and the score of the nodes casting those votes. The sentences are ranked based on this score. Top ranked sentences are selected for summary based on summary length. The authors

Table 1. Terms notations

Term	Description	Term	Description
Doc	Document	IS	Isolated Sentences
AVG	Average	β – beta	P/R
NOS	Number Of Sentences	ROUGE-1	Unigram based co-occurrence statistics
X	System summary	ROUGE -2	Bigram based co-occurrence statistics
Y	Human summary	ROUGE –L	LCS based co-occurrence statistics
LCS	Longest Common Subsequence	R_1	ROUGE-1 recall
LCS(X,Y)	Length of LCS between X and Y	R_2	ROUGE -2 recall
SL	Summary Length	R_L	ROUGE-L recall

Table 2. Dataset statistics

Argument	Value	Argument	Value
Number of Documents	10	Summary as % of Document Length	30%
AVG NOS per Document	23	AVG Summary size in NOS	7
Maximum NOS per Document	32	Maximum NOS per Summary	10
Minimum NOS per Document	14	Minimum NOS per Summary	3
AVG Number of IS per Document	2	Number of human Summaries per Document	1

performed two kinds of evaluation: intrinsic content based evaluation and graph-structure based evaluation.

Content Based Evaluation

The authors used ROUGE metric for content based evaluation (Saggion & Poibeau, 2016; Kiyoumars, 2014; Lin, 2004; Lin & Hovy, 2003). It measures the similarity between system generated summary and human summary. The ROUGE is found to be highly correlated with human evaluations. It includes the evaluation methods ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S, and ROUGE-SU. Here, the authors performed ROUGE-N and ROUGE-L evaluations. The ROUGE-N is an N-gram based statistics. The N-gram[1] may be Unigram[2], Bigram[3], Trigram[4], and so on. The authors used rouge version 0.3.1 which is available at https://pypi.org.project/rouge/ for summary evaluation. Table 3 contains the formulas for ROUGE-1 and ROUGE-L calculations.

The authors computed ROUGE-1, ROUGE-2 and ROUGE-L (summary level) on the document summaries of the MCBSS approach and the existing graph approaches BP, AS, and TR and compared them. Compared to ROUGE-2, ROUGE-1 and ROUGE-L correlate highly with human judgment. Since ROUGE is a recall oriented metric, the authors consider only the recalls of all the summaries for comparison. Table 4 shows the ROUGE-1 recall of each summary of ten documents. Figure 5 shows the bar chart of Table 4. Refer to Table 4 and Figure 5; it is observed that the MCBSS summary has high ROUGE-1 recall than other summaries for all documents except Doc 2, Doc 4 and Doc 10. The average ROUGE-1 score of the MCBSS summary is also higher than other summaries.

Table 5 shows the ROUGE-L recall of each summary of ten documents. Figure 6 shows the corresponding bar chart. Refer to Table 5 and Figure 6; it is observed that the MCBSS summarizer performs better than other summarizers for 50% of documents. Its average ROUGE-L score is also higher than other summarizers. From Tables 4 and 5, it is observed that the ROUGE-L score is always less than or equal to the ROUGE-1 score. From ROUGE-1 and ROUGE-L recalls, it is clearly understood that the MCBSS summary correlates highly with human summary than other summaries. Hence, the MCBSS summarizer produces an informative summary.

Graph-Structure Based Evaluation

It is a new evaluation scheme proposed in this article. In graph-structure based evaluation, the authors used the MCs of the text relationship graph to compare the MCBSS summary with the summaries of BP, AS, and TR. Table 6 shows the required statistics of each input document and its summaries. Table 7 shows the input text documents and their associated list of MCs.

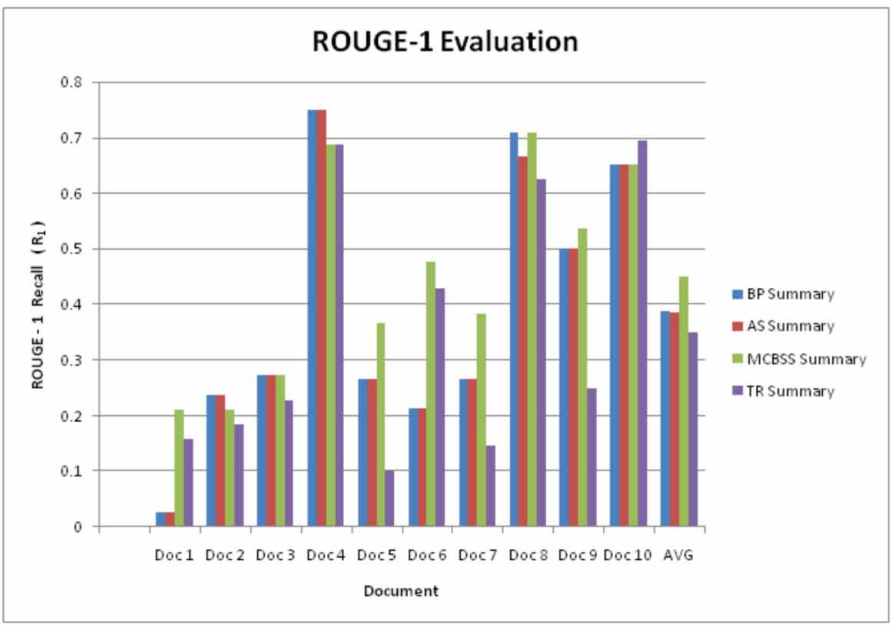
Table 3. ROUGE score calculation

Metric	ROUGE-1 Formula	ROUGE-L Formula
Precision (P)	$\frac{\lvert Common\,Unigrams\,in\,X\,and\,Y \rvert}{\lvert Unigrams\,in\,X \rvert}$	$\frac{\lvert LCS(X,Y) \rvert}{\lvert X \rvert}$
Recall (R)	$\frac{\lvert Common\,Unigrams\,in\,X\,and\,Y \rvert}{\lvert Unigrams\,in\,Y \rvert}$	$\frac{\lvert LCS(X,Y) \rvert}{\lvert Y \rvert}$
F-measure	$\frac{2 * P * R}{P + R}$	$\frac{(1+\beta^2) * R * P}{R + (\beta^2 * P)}$

Table 4. ROUGE-1 Recall

Doc No	BP Summary R_1	AS Summary R_1	MCBSS Summary R_1	TR Summary R_1
Doc 1	0.0263	0.0263	0.2105	0.1579
Doc 2	0.2368	0.2368	0.2105	0.1842
Doc 3	0.2727	0.2727	0.2727	0.2272
Doc 4	0.75	0.75	0.6875	0.6875
Doc 5	0.2667	0.2667	0.3667	0.1
Doc 6	0.2143	0.2143	0.4762	0.4286
Doc 7	0.2647	0.2647	0.3824	0.1471
Doc 8	0.7083	0.6667	0.7083	0.625
Doc 9	0.5	0.5	0.5357	0.25
Doc 10	0.6522	0.6522	0.6522	0.6957
AVG	0.3892	0.3850	0.4503	0.3503

Figure 5. ROUGE-1 recall

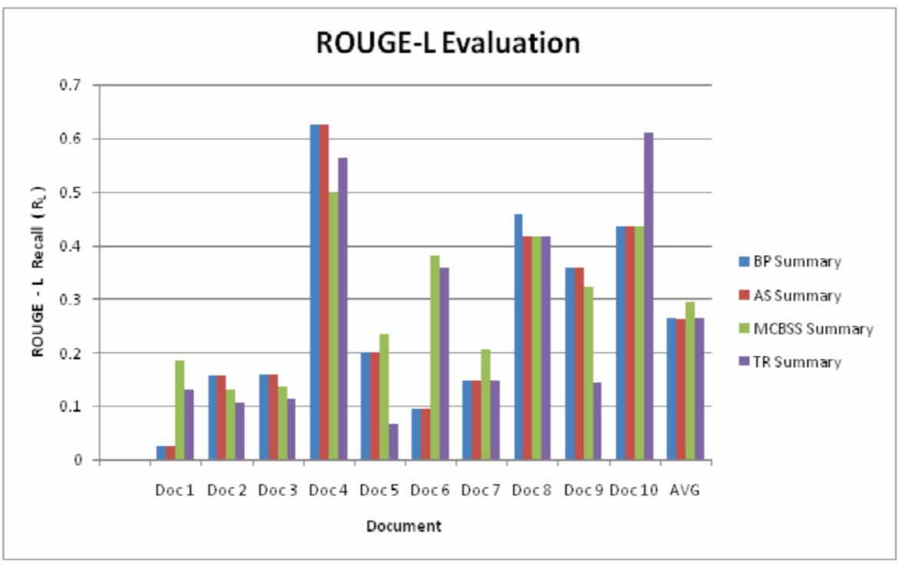


From Table 6, it is clearly understood that both BP and AS select same set of sentences, only few of them are varied. Also in both approaches, most of the summary sentences for each document are in the same MC (see Table 6 and Table 7) and are adjacent. So they convey almost similar information with little difference. Hence, they introduce redundancy. But there are a lot of important sentences in other MCs which should be selected. So coverage is a problem here. Thus, the produced summary does not contain enough information to understand the original text.

Table 5. Rouge-L Recall

Doc No	BP Summary R_L	AS Summary R_L	MCBSS Summary R_L	TR Summary R_L
Doc 1	0.0263	0.0263	0.1842	0.1316
Doc 2	0.1579	0.1579	0.1316	0.1053
Doc 3	0.1591	0.1591	0.1364	0.1136
Doc 4	0.625	0.625	0.5	0.5625
Doc 5	0.2	0.2	0.2333	0.0667
Doc 6	0.0952	0.0952	0.3810	0.3571
Doc 7	0.1471	0.1471	0.2059	0.1471
Doc 8	0.4583	0.4167	0.4167	0.4167
Doc 9	0.3571	0.3571	0.3214	0.1429
Doc 10	0.4348	0.4348	0.4348	0.6087
AVG	0.2661	0.2619	0.2945	0.2652

Figure 6. ROUGE-L recall



TR identifies important sentences in a text using the text rank score of the sentences (Mihalcea & Tarau, 2004). Although it is the best graph ranking algorithm, it does not guarantee that the selected sentences are disjointed (see Table 6 and Table 7). Also, there is no chance for selecting isolated sentences; because of no connectivity they always have a low text rank score. Hence, it does not provide diverse or novel information.

The MCBSS algorithm covers all concepts of the input text by selecting summary sentences in distinct MCs (see Table 6 and Table 7). So the summary sentences are not adjacent. Since adjacent sentences share similar information, it avoids redundancy by omitting adjacent sentences. Thus, the

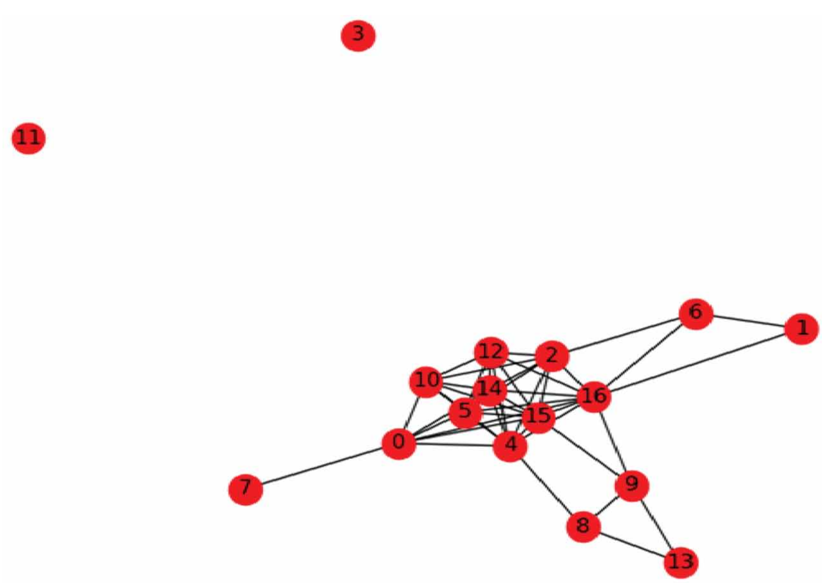
Table 6. Documents statistics and their system summaries

Doc No	NOS	SL	IS	BP Summary	AS Summary	MCBSS Summary	TR Summary
Doc 1	17	5	3,11	2,4,5,15,16	4,5,14,15,16	0,1,2,3,8	0,6,7,10,15,16
Doc 2	27	8	17	0, 5, 7, 8, 18, 19, 23, 26	0, 5, 6, 7, 14, 18, 19, 26	0, 1, 9, 11, 17, 19, 22, 24	0,5,11
Doc 3	32	10	-	0, 1, 4, 5, 6, 21, 25, 27, 29, 30	0, 1, 4, 5, 6, 21, 25, 27, 29, 30	0, 7, 15, 17, 19, 25	0,4,18
Doc 4	20	6	18,19	0, 1, 2, 6, 9, 17	0, 1, 6, 8, 9, 17	0, 10, 13, 16, 18	0,5
Doc 5	29	9	28	0, 1, 3, 8, 12, 18, 20,23, 24	0, 1, 3, 8, 12, 18, 20, 23, 24	0, 21, 2, 19, 15, 7, 10, 11,28	14,18
Doc 6	14	4	3,11,13	2, 5, 7, 9	2, 5, 7, 9	0, 3, 5, 8	0,2
Doc 7	25	8	16	0, 1, 3, 4, 10, 13, 14, 17	0, 1, 4, 10, 11, 13, 14, 17	0, 6, 8, 12, 15, 16, 18, 19	0,14
Doc 8	29	9	25	0, 1, 5, 7, 10, 12, 13, 16, 20	0, 1, 5, 7, 10, 13, 15, 16, 20	0, 2, 3, 10, 11, 22, 25, 27	0,1,3
Doc 9	18	3	12,16,17	0,2,9,13,15	0,2,9,13,15	0,5,14,17	0,9,5
Doc 10	22	7	14,18	0,2,5,6,7,11,17	0, 2, 5, 6, 7, 11, 17	0, 8, 13, 14, 15, 21	0,1

MCBSS approach provides coverage and non-redundancy. Also, it provides novelty by selecting isolated sentences and utilizes group relations using the logical structures (MCs) of the graph.

The authors explain the above findings with one example. Consider Doc 1 in Table 6. Figure 7 shows the sentence graph of Doc1. BP selects the sentences 2, 4, 5, 15, and 16 of Doc 1. The AS selects the sentences 4, 5, 14, 15, and 16 of Doc 1. Both select the sentences 4, 5, 15, and 16. But the sentences 4, 5, 15, and 16 are in the same MC of Doc 1 in Table 8. So they convey similar information. TR selects sentences 0, 6, 7, 10, 15, and 16 of Doc 1. But the sentences 0, 15, and 16

Figure 7. Sentence graph of document 1



are in the same MC of Doc1 in Table 8. So TR selects adjacent sentences. Hence, TR introduces redundancy. Moreover from Figure 7 and Table 6, it is easy to understand that the approaches BP, AS, and TR never select the isolated sentences 3 and 11. So they fail to provide novel information.

The MCBSS algorithm selects sentences 0, 1, 2, 3, and 8 of Doc 1 in Table 6. All these sentences are in distinct maximal cliques of Doc 1 in Table 7, so they are non-redundant. The sentences 0, 1, 2, and 8 are enough to cover all MCs of Doc 1 in Table 7. Hence, it covers all concepts of the entire text. Since it contains the isolated sentence 3, it provides novelty. Thus, the MCBSS summary provides non-redundant and novel sentences that cover all concepts of the input text.

Table 7. Documents and their list of maximal cliques in their sentence graphs

Doc No.	Maximal Cliques (size >=2)
Doc 1	[[7, 0], [8, 9, 13], [8, 4], [10, 4, 5, 14, 15, 0], [10, 4, 5, 14, 15, 2, 12], [16, 1, 6], [16, 6, 2], [16, 15, 9], [16, 15, 4, 5, 14, 0], [16, 15, 4, 5, 14, 2, 12]]
Doc 2	[[3, 18, 19], [4, 19], [5, 0, 15], [5, 0, 7, 8, 18], [5, 0, 7, 6], [5, 1, 8], [5, 2], [5, 9, 18, 7], [5, 11, 6, 14], [5, 22], [10, 8, 23, 1], [10, 8, 23, 18, 19, 7], [12, 1, 13], [13, 14], [16, 2], [16, 18, 0], [16, 18, 9], [16, 22], [19, 25, 26], [19, 2], [19, 7, 26, 18], [20, 25, 26], [20, 6, 11, 26], [20, 6, 11, 14], [20, 22], [21, 18, 0], [21, 18, 26], [23, 11, 14], [23, 22], [24, 26], [26, 6, 7]]
Doc 3	[[1, 16, 2, 26], [1, 16, 2, 5, 0], [1, 16, 2, 5, 29], [1, 16, 18, 0], [1, 18, 0, 21, 6], [1, 18, 19], [1, 30, 2, 21, 26], [1, 30, 2, 21, 5, 0], [1, 30, 2, 21, 5, 29], [1, 30, 27, 29, 25], [1, 30, 27, 29, 5, 19], [1, 30, 27, 29, 5, 21], [1, 30, 27, 5, 0, 21], [1, 30, 27, 6, 24, 0], [1, 30, 27, 6, 24, 25, 26], [1, 30, 27, 6, 21, 0], [1, 30, 27, 6, 21, 26], [1, 30, 23, 7], [1, 30, 7, 6], [3, 20, 9], [3, 20, 14, 15], [4, 16, 2, 26], [4, 16, 2, 5, 0], [4, 16, 2, 5, 29], [4, 2, 21, 26], [4, 2, 21, 5, 0], [4, 2, 21, 5, 29], [4, 27, 25, 26], [4, 27, 25, 31, 29], [4, 27, 25, 31, 22], [4, 27, 26, 21], [4, 27, 5, 21, 0], [4, 27, 5, 21, 29], [4, 27, 5, 22], [4, 12, 25, 26], [8, 6], [8, 15], [9, 0, 16], [9, 29, 16], [9, 29, 19, 20], [10, 5, 11, 21], [10, 5, 19], [11, 15], [13, 16, 0], [13, 16, 29], [14, 0], [17, 16, 2], [20, 19, 18], [20, 19, 30, 29], [22, 24, 25, 27, 31], [22, 28, 25, 31], [31, 25, 27, 30, 24], [31, 25, 27, 30, 29]]
Doc 4	[[9, 17, 0, 4, 8, 1, 2], [9, 17, 0, 6, 15, 11], [9, 17, 0, 6, 7, 1, 8, 2], [9, 17, 0, 6, 7, 1, 11], [9, 17, 0, 6, 7, 3, 8, 2], [9, 17, 16, 11, 1], [9, 17, 16, 11, 15], [9, 17, 13, 11, 15], [9, 17, 13, 4], [9, 10, 11, 7], [9, 5, 0, 2, 8, 1, 4], [9, 5, 0, 2, 8, 1, 6, 7], [9, 5, 0, 2, 8, 3, 6, 7], [9, 14, 16], [9, 14, 4, 13], [12, 0, 1, 2, 8, 5, 4], [12, 0, 1, 2, 8, 5, 6], [12, 0, 1, 11, 6], [12, 0, 15, 11, 6]]
Doc 5	[[6, 8], [6, 19], [7, 16], [7, 17], [9, 10], [9, 18, 1, 14], [9, 18, 1, 15], [9, 18, 2], [11, 1], [12, 8, 4, 0, 24], [12, 8, 23, 20, 0, 24], [12, 8, 23, 20, 0, 1, 17, 18, 3], [12, 8, 23, 20, 2, 24], [12, 8, 23, 20, 2, 18], [12, 8, 23, 20, 21, 1, 3], [12, 8, 23, 5, 24, 2], [12, 19, 24, 5, 22], [12, 22, 1, 20], [12, 22, 2, 24, 20], [12, 22, 2, 24, 5], [13, 0, 25, 26, 27], [14, 24, 19], [14, 1, 3, 18], [15, 1, 3, 18, 23], [15, 1, 22]]
Doc 6	[[0, 2, 4, 7], [0, 12], [5, 6], [5, 7, 2, 1, 9], [5, 7, 2, 4], [5, 7, 10], [8, 9], [8, 4], [12, 1, 9]]
Doc 7	[[0, 2, 4, 1, 9], [0, 2, 4, 13], [0, 10, 17, 9, 1], [0, 10, 17, 9, 11], [0, 10, 17, 11, 13], [0, 10, 17, 14, 1], [0, 10, 17, 14, 3], [0, 10, 17, 14, 13], [0, 10, 4, 1, 9], [0, 10, 4, 1, 14], [0, 10, 4, 11, 9], [0, 10, 4, 11, 13], [0, 10, 4, 13, 14], [0, 10, 5, 1], [0, 10, 5, 3], [0, 22, 3], [6, 2], [6, 3], [6, 21, 7], [7, 12], [7, 4, 10], [8, 20], [15, 13], [18, 14], [18, 23], [19, 21], [21, 24], [23, 1], [23, 22]]
Doc 8	[[2, 1, 18], [2, 1, 7], [2, 19], [2, 4, 6], [4, 17], [4, 27], [4, 12, 6], [4, 13], [6, 12, 20, 22], [11, 9, 23], [11, 14, 7], [14, 1, 13, 7, 0], [14, 1, 13, 7, 10], [14, 17, 22], [16, 0, 24, 26], [16, 0, 24, 15, 7], [16, 0, 1, 26], [16, 0, 1, 13, 15, 5], [16, 0, 1, 13, 15, 7], [16, 24, 28, 7], [16, 10, 5, 20, 28], [16, 10, 5, 13, 8, 28], [16, 10, 5, 13, 1, 23, 15], [16, 10, 7, 20, 28], [16, 10, 7, 13, 1, 15], [16, 10, 7, 13, 28], [16, 10, 9, 12, 8], [16, 10, 9, 12, 26], [16, 10, 9, 23], [16, 10, 12, 20, 26], [16, 10, 26, 1], [16, 3, 12, 8], [16, 3, 12, 20], [16, 3, 5, 8], [16, 3, 5, 20], [16, 3, 5, 23], [18, 0, 1], [18, 12, 22], [21, 10, 1], [21, 10, 20], [21, 22, 20], [22, 8, 12]]
Doc 9	[[0, 1, 4], [0, 10], [0, 11, 2, 4, 13, 15], [0, 9, 8, 2, 7, 15], [0, 9, 13, 3, 6], [0, 9, 13, 15, 2, 4], [0, 9, 13, 15, 2, 7], [0, 9, 13, 15, 6], [5, 1], [5, 11, 13, 15], [14, 9, 6]]
Doc 10	[[5, 8], [5, 16, 17, 10, 11, 15], [5, 6, 0, 2, 4, 11, 17, 1, 7], [5, 6, 0, 2, 4, 11, 17, 12, 10], [5, 6, 0, 2, 4, 11, 17, 12, 7], [5, 6, 0, 3, 7, 1], [5, 6, 0, 3, 7, 12], [5, 6, 15, 2, 11, 17, 1, 7], [5, 6, 15, 2, 11, 17, 10], [9, 8], [13, 12, 7], [19, 21], [20, 21]]

CONCLUSION

The main goal of this article is the generation of the non-redundant, novel, and informative summary that covers all concepts of the input document using the MCBSS algorithm. The authors proved that the MCBSS summarizer produces an informative summary through content based evaluation. Also, they proved that the MCBSS summarizer provides non-redundant and novel summary sentences that cover all concepts of the input text through graph-structure based evaluation.

The benefits of the proposed approach are: (i) there is no need for checking redundancy because the selected sentences are disjoint; (ii) since each summary sentence discusses topics covered by many other sentences which are not selected, the summary is highly informative ; (iii) since it selects one sentence from each concept of the text, the coverage of the text is good; (iv) it is easy to find novel information using isolated sentences; (v) since MCs form natural clustering of sentences, it requires no explicit mechanism for clustering sentences. It simply selects summary sentences from the largest size MCs (vi) this approach could be suitable for multi-document summarization, because there is high degree of redundancy; (vii) the generated summary would be suitable for educational domain because informativeness, non-redundancy, novelty, and concept coverage are the main aspects of the summary of any study-material.

Since the proposed work selects summary sentences from each concept of the input text, it gives an overall idea of all concepts of the text. Thus, it provides global coherence. It does not guarantee for local coherence because the selected sentences are disjoint. So the future work will aim to improve local coherence. In the future, this approach can be extended to multiple documents for domain specific or topic specific summary generation.

REFERENCES

Aries, A., Zegour, D. E., & Hidouci, W. K. (2018). Exploring graph bushy paths to improve statistical multilingual automatic text summarization. *International Federation for Information Processing*, 78-89.

Calvo, H., Carrillo-Mendoza, P., & Gelbukh, A. (2018). On redundancy in multi document summarization. *Journal of Intelligent & Fuzzy Systems*, 1–11.

Chen, H., & Zhao, H. J. (2014). PSG: A two-layer graph model for document summarization. *Frontiers of Computer Science*, *8*(1), 119–130. doi:10.1007/s11704-013-2292-2

Erkan, G., & Radev, D. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479. doi:10.1613/jair.1523

Fattah, M. A., & Ren, F. (2009). Automatic text summarization. *International Journal of Computational Science*, *3*(1), 25–28.

Feiyue, Y., & Xinchen, X. (2018). Automatic multi-document summarization based on keyword density and sentence-word Graphs. *J. Shanghai Jiao Tong*. *Universitas Scientiarum*, *23*(4), 584–592. doi:10.1007/s12204-018-1957-2

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

Ferreira, R., Freitas, F., Cabral, L. S., Lins, R. D., Lima, R., Franc, G., . . . Favaro, L. (2013b). A four dimension graph model for automatic text summarization. In *Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology*. IEEE Computer Society. doi:10.1109/WI-IAT.2013.55

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and techniques*. Morgan Kaufmann Publishers.

Han, X., Lv, T., Hu, Z., Wang, X., & Wang, C. (2016). Text summarization using framenet based semantic graph model. *Hindawi Publishing Corporation Scientific Programming*, *2016*, 1–10. doi:10.1155/2016/5130603

John, A., & Wilscy, M. (2015). Vertex cover algorithm based multi-document summarization using information content of sentences. In *Proceedings of International Conference on Information and Communication Technologies*. Procedia Computer Science. doi:10.1016/j.procs.2015.02.022

Khushboo, S., Dharaskar, R. V., & Chandak, M. B. (2010). Graph-based algorithms for text summarization. In *Proceedings of Third International Conference on Emerging Trends in Engineering and Technology*. IEEE.

Kiyoumarsi, F. (2014). Evaluation of automatic text summarizations based on human summaries. In *Proceedings of Second Global Conference on Linguistics and Foreign Language Teaching*. Procedia - Social and Behavioral Sciences.

Lin, C. Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization*. Barcelona, Spain: Association for Computational Linguistics.

Lin, C. Y., & Hovy, E. H. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. Edmonton, Canada: Association for Computational Linguistics. doi:10.3115/1073445.1073465

Luhn, H. P. (1958). *The automatic creation of literature abstracts.* Paper presented at IRE National Convention, New York, NY.

Mani, I., & Bloedorn, E. (1997). Multi-document summarization by graph search and matching. *Natural Language Generation*, 622-628.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain: Association for Computational Linguistics.

Mihalcea, R., & Tarau, P. (2005). A language independent algorithm for single and multiple document summarization. *Proceedings of the Second International Joint Conference on Natural Language Processing*.

Nenkova, A., & McKeown, K. (2011). Automatic summarization. *Foundations and Trends in Information Retrieval*, *5*(2-3), 103–233. doi:10.1561/1500000015

Oliveira, H., Ferreira, R., Lima, R., Lins, R. D., Freitas, F., Riss, M., & Simske, S. J. (2016). Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. *Expert Systems with Applications*, *65*, 68–86. doi:10.1016/j.eswa.2016.08.030

Saggion, H., & Poibeau, T. (2016). Automatic text summarization: Past, present and future. In T. Poibeau, H. Saggion, J. Piskorski, & R. Yangarber (Eds.), *Multi-source, Multilingual Information Extraction and Summarization* (pp. 3–22). Springer.

Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). Automatic text structuring and summarization. *Information Processing & Management*, *33*(2), 193–207. doi:10.1016/S0306-4573(96)00062-3

Sariki, T. P., Kumar, G. B., Shukla, U., & Mishra, A. (2019). An adroit approach for extractive text summarization. *International Journal of Engineering and Advanced Technology, 8*(5), 2047-2051.

School of Computing Science. (2013). Review of the Bron-Kerbosch algorithm and variations. University of Glasgow.

Shen, C., & Li, T. (2010). Multi-Document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Beijing, China: Association for Computational Linguistics.

Simon, J. R., Ledeneva, Y., & Garcıa-Hernandez, R. A. (2018). Calculating the significance of automatic extractive text summarization using a genetic algorithm. Journal of Intelligent and Fuzzy System. doi:10.3233/JIFS-169588

Sornil, O., & Gree-ut, K. (2006). *An automatic text summarization approach using content-based and graph-based characteristics*. CIS-IEEE. doi:10.1109/ICCIS.2006.252361

Steinberger, J., & Jezek, K. (2009). Evaluation measures for text summarization. *Computer Information*, *28*, 1001–1026.

Takamura, H., & Okumura, M. (2009). Text Summarization Model based on Maximum Coverage Problem and its Variant. In *Proceedings of the 12th Conference of the European Chapter of the ACL*. Greece, Athens: Association for computational linguistics. doi:10.3115/1609067.1609154

Tixier, A., Meladianos, P., & Vazirgiannis, M. (2017). Combining graph degeneracy and sub modularity for unsupervised extractive summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. Copenhagen, Denmark: ACL. doi:10.18653/v1/W17-4507

Tomita, E., Akutsu, T., & Matsunaga, T. (2011). Efficient algorithms for finding maximum and maximal cliques: Effective tools for bioinformatics. *Biomedical Engineering, Trends in Electronics, Communications and Software,* 625-641.

Verma, P., & Om, H. (2019). A novel approach for text summarization using optimal combination of sentence scoring methods. *Sadhana*, *44*(5), 1–15. doi:10.1007/s12046-019-1082-4

Yang, K., Al-Sabahi, K., Xiang, Y., & Zhang, Z. (2018). An integrated graph model for document summarization. *Information, 9*, 232, 1-13.

Zheng, H. T., & Bai, S. Z. (2014). Graph-Based Summarization without Redundancy. In L. Chen, Y. Jia, T. Sellis, & G. Liu (Eds.), *Web Technologies and Applications* (pp. 449–460). Springer. doi:10.1007/978-3-319-11116-2_39

ENDNOTES

[1] A subsequence of n words from a text.
[2] Independent words of a given text.
[3] A subsequence of two words from a text.
[4] A subsequence of three words from a text.

Krishnaveni P. received the M.E. degree in Computer Science and Engineering from Anna University, Tamil Nadu. She is currently pursuing the Ph.D. degree in the Department of Computer Applications, National Institute of Technology, Tamil Nadu. Her research interests include Information retrieval, Natural language processing, and Automatic text summarization.

S. R. Balasundaram has been working since 1987 at National Institute of Technology, Tiruchirappalli (NITT). He obtained M.C.A. degree from P.S.G. College of Technology, Coimbatore and M.E. in Computer Science and Engineering from Regional Engineering College, Tiruchirappalli. He earned his doctorate in "E-Learning and Assessment" from NITT. Currently, he is working as Professor in the Department of Computer Applications, NITT. He has more than 40 papers in reputed Journals and proceedings of International Conferences. His areas of interest are: Web and Mobile Technologies, Cognitive Science and e-Learning Technologies. He has authored a book, co-author for 3 book chapters, and edited a book.