GUSUM: Graph-Based Unsupervised Summarization using Sentence Features Scoring and Sentence-BERT

Tuba Gokhan, Phillip Smith and **Mark Lee**
School of Computer Science
University of Birmingham, United Kingdom
{txg857|smithpm|m.g.lee}@cs.bham.ac.uk

Abstract

Unsupervised extractive document summarization aims to extract salient sentences from a document without requiring a labelled corpus. In existing graph-based methods, vertex and edge weights are usually created by calculating sentence similarities. In this paper, we develop a Graph-Based Unsupervised Summarization(GUSUM) method for extractive text summarization based on the principle of including the most important sentences while excluding sentences with similar meanings in the summary. We modify traditional graph ranking algorithms with recent sentence embedding models and sentence features and modify how sentence centrality is computed. We first define the sentence feature scores represented at the vertices, indicating the importance of each sentence in the document. After this stage, we use Sentence-BERT for obtaining sentence embeddings to better capture the sentence meaning. In this way, we define the edges of a graph where semantic similarities are represented. Next we create an undirected graph that includes sentence significance and similarities between sentences. In the last stage, we determine the most important sentences in the document with the ranking method we suggested on the graph created. Experiments on CNN/Daily Mail, New York Times, arXiv, and PubMed datasets show our approach achieves high performance on unsupervised graph-based summarization when evaluated both automatically and by humans.

1 Introduction

Text summarization is the process of compressing a long text into a shorter version while preserving key information and significance of the content. Researchers have examined two summarization models as *extractive* and *abstractive* summarization (Nenkova et al., 2011). Extractive summarization creates summaries by extracting text from source documents, whereas abstractive summarization rewrites documents by paraphrasing or deleting some words or phrases.

Modern text summarization approaches focus on supervised neural networks, which adapt sequence-to-sequence translation, reinforcement learning and large-scale pre-training techniques. These approaches have accomplished favourable results thanks to the availability of large-scale datasets (Nallapati et al., 2016; Cheng and Lapata, 2016; Gehrmann et al., 2018; Liu and Lapata, 2019; Wang et al., 2020). Nevertheless, a major limitation of those supervised methods is that their success is strongly reliant on the availability of large training corpora with human-generated high-quality summaries which are both expensive to produce and difficult to obtain. We focus on unsupervised summarization in this study, where we simply need unlabeled documents.

The fundamental issue with unsupervised summarizing is determining which sentences in a document are important. Graph-based algorithms, in which each vertex is a sentence and the weights of the edges are measured by sentence similarity, are the most prevalent approaches among these studies. The relevance of each sentence is then estimated using a graph ranking approach. A vertex's *centrality* is often measured using graph-based ranking algorithms such as PageRank (Brin and Page, 1998) to decide which sentence to include in the summary.

We observe that the importance of the sentences in the document should be emphasized in addition to the semantic similarity of the sentences in the summary. Accordingly, we suggest in this study that the centrality measure can be enhanced in two significant ways. First, we define an initial score that specifies the importance of the sentence that each vertex represents. Second, we use Sentence-BERT (Reimers and Gurevych, 2019) which is a modification of the pre-trained BERT network (Devlin et al., 2019) that uses Siamese and triplet network structures to derive semantically meaningful sentence embeddings to better capture the sentence

Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing, pages 44–53
October 16, 2022.

meaning and calculate sentence similarity.

In this paper, we propose a novel approach, GUSUM (as shorthand for **G**raph-Based **U**nsupervised **Sum**marization) which is a simple and powerful approach to improving graph-based unsupervised extractive text summarization. We evaluate the GUSUM on the CNN/Daily Mail and New York Times short document summarization datasets and arXiv and PubMed long document summarization datasets. For graph-based summarization tasks, pre-trained embeddings are generally used only for measuring sentence similarities in graph-based summarization systems. However, this situation causes the importance of the sentences in the document to be ignored. In our approach, we applied a ranking method that combines sentence similarities and sentence features to calculate sentence centrality. Our experiments show that better results are obtained by creating weighted graphs in which the main features of the sentence are represented in the ordering stage based on sentence centrality. Our code is available at https://github.com/tubagokhan/GUSUM

2 Related Work

The proposed method is based on graph-based, unsupervised extractive text summarization techniques. In this section, we introduce work on graph-based summarization, unsupervised summarization and pre-training.

2.1 Graph-Based Unsupervised Summarization

The majority of summarization methods rely on labeled datasets containing documents that match pre-prepared summaries. Compared to supervised models, unsupervised models only need unlabeled documents during training. Most unsupervised extractive models are graph-based (Carbonell and Goldstein, 1998; Erkan and Radev, 2004; Mihalcea and Tarau, 2004; Zheng and Lapata, 2019; Xu et al., 2020; Liang et al., 2021; Liu et al., 2021). Among the representative examples of early work in inferential summarization, the study by Carbonell and Goldstein (1998) includes the Maximum-Marginal Relevance (MMR) principle of selecting sentences based on both the relevance and diversity of the selected sentences and the PageRank (Brin and Page, 1998) scores of the sentences in sentence similarity graphs. TEXTRANK (Mihalcea and Tarau, 2004) interprets sentences in a document as nodes in an undirected graph, with edge weights based on sentence occurrence similarity. The final ranking scores for sentences are then determined using graph-based ranking algorithms such as PageRank. Similarly, Erkan and Radev (2004) provided extractive summaries by scoring sentences with the LEXRANK approach, they calculated the importance of sentences in representative graphs based on the measurement of eigenvector centrality.

Recently, researchers have continued to develop graph-based methods. Zheng and Lapata (2019) created a directed graph using BERT (Devlin et al., 2019) to calculate sentence similarities. The importance score of a sentence is the weighted sum of all its outer edges, where weights for edges between the current sentence and preceding sentences are negative. In the directed graph that Zheng and Lapata (2019) created, the edges represent the relative position of the sentences in the document. In our study, we represented sentence similarities at the edges from a completely different point of view. We also showed vertexes by blending the features of the sentences such as the position of the sentence. Thus, we created graphs that provide greater semantic integrity. Xu et al. (2020) design two summarization tasks related to pre-training tasks to improve sentence representation. Then they proposed a rank method that combines attention weight with reconstruction loss to measure the centrality of sentences. Liang et al. (2021) proposed a facet-sensitive centrality-based model. It aims to measure the relationship between the summary and the document by calculating a similarity score between the summary sentences and the document for each candidate summary. Liu et al. (2021) published a graph-based single-document unsupervised extractive method that constructs a Distance-Augmented Sentence Graph from a document that enables the model to perform more fine-grained modeling of sentences and better characterize the original document structures.

2.2 Pre-trained Language Models

Pre-trained language models have been shown to make significant progress in a variety of NLP tasks. These models are based on the concept of word embeddings (Pennington et al., 2014), but they go even further by pre-training a sentence encoder on a large unlabeled corpus. Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), one of the state-of-art language

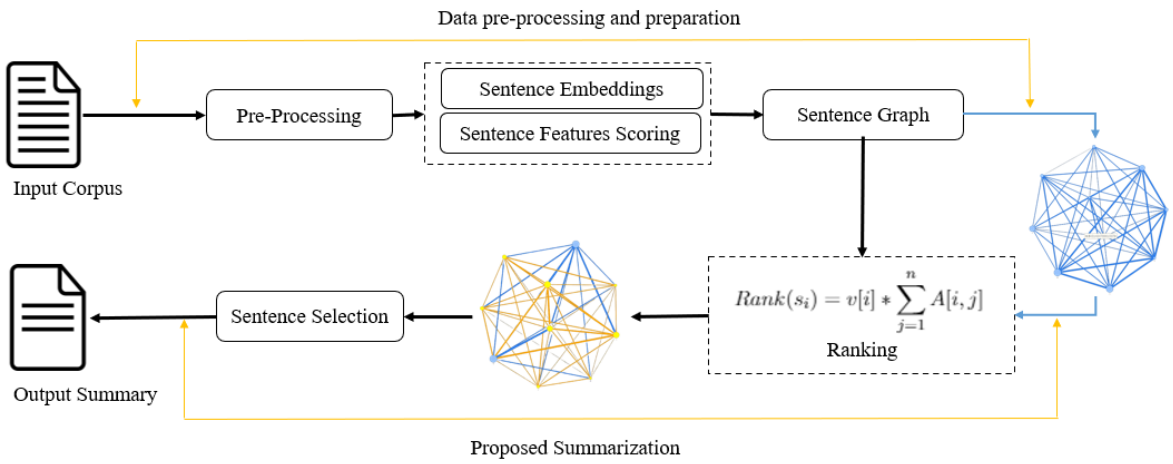

Figure 1: The complete pipeline of the proposed method.

models, is trained with a masked language model and a next-sentence-predicting task. Pre-trained language models have recently become popular for improving performance in language comprehension tasks. Recent research (Liu and Lapata, 2019; Bae et al., 2019) has shown that using pre-trained language models to extractive summarization models, such as BERT, is quite advantageous. As for the extractive summarization task, it provides the powerful sentence embeddings and the contextualized information among sentences (Zhong et al., 2019), which have been proven to be critical to extractive summarization.

3 Methodology

In this section, we describe our unsupervised summarization method GUSUM. The system is composed of four main steps: first, we calculate sentence features for defining vertex weight; second, we produce sentence embeddings by Sentence-BERT to measure sentence similarities; next, we create a graph by comparing all the pairs of sentence embeddings obtained; finally, we rank the sentences by their degree centrality in this graph. Figure 1 gives an overview of the whole proposed method.

3.1 Computing Sentence Features

In traditional embedding-based systems, sentence features are transformed into dense vector representation. These features are attributes that attempt to represent the data used for their task (Suanmali et al., 2009).

Unlike traditional methods, GUSUM uses sentence features to determine the initial rank of the vertex in the generated graphs rather than vectorizing them. GUSUM focuses on four features for each sentence based on Shirwandhar and Kulkarni (2018). After the scores for each sentence were determined, the sum of the scores was assigned by taking the weight of the vertex representing the sentence.

Sentence length: This feature is useful for filtering out short phrases commonly found in news articles, such as dates and author names. Short sentences do not contain much information and are not expected to belong to the summary. To find the important sentence based on its length, the feature score is calculated using 1:

$$Score_{f1}(S_i) = \frac{No.\,Word\,in\,S_i}{No.Word\,in\,Longest\,Sentence} \quad (1)$$

Sentence position: On the basis of sentence position, its relevance is known. The first and the last sentence of a document are typically important and involve maximum information. Position feature is calculated using 2:

$$Score_{f2}(S_i) = \begin{cases} 1 & if\ the\ first\ or\ last\ sentence \\ \frac{N-P}{N} & if\ others \end{cases} \quad (2)$$

where, N is the total number of sentences and P is the position of the sentence.

Proper nouns: Usually, the sentence that contains more proper nouns is an important one and it is most probably included in the document summary. The score for this feature is calculated as the ratio of the number of proper nouns in a sentence over the sentence length using a POS tagger as in 3.

$$Score_{f3}(S_i) = \frac{No.\,Proper\,Noun\,in\,S_i}{Length\,S_i} \quad (3)$$

Numerical token: The number of numerical tokens that present in the sentence is another feature that shows the importance of the sentence in the document and is calculated with 4:

$$Score_{f4}(S_i) = \frac{num\_numeric_i}{Length\, S_i} \quad (4)$$

where, $num\_numeric_i$ is the total number of numerical tokens in sentence i.

3.2 Computing Sentence Embeddings

The first step in our pipeline is to generate a list of sentences from the compilation text. After extracting the sentences, the next step is to produce the sentence embedding of each sentence using Sentence-BERT (Reimers and Gurevych, 2019). Sentence-BERT is a modification of the pre-trained BERT (Devlin et al., 2019) network that uses Siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using vector similarity methods.

The proposed approach uses Sentence-BERT[1] embeddings to represent sentences as fixed-size vectors. Thus, all sentences and the source is mapped in the same semantic space and taken as inputs to the system.

3.3 Generation of the Sentence Graph

In our unsupervised graph-based extractive summarization approach, the document is represented as a graph, where each node represents a sentence in the input document.

Given a document D, it contains a set of sentences $(s_1, s_2, ..., s_n)$. Graph-based algorithms treats D as a graph $G = (V; E)$. $V = (v_1, v_2, ..., v_n)$ is the vertex set where v_i is the representation of sentence s_i. E is the edge set, which is an $n \times n$ matrix. Each $= e_{i,j} \in E$ denotes the weight between vertex v_i and v_j.

In graph-based summarization methods, centrality is used to select the most salient sentence to construct summaries through ranking. Centrality of a node measures its importance within a graph. The key idea of graph-based ranking is to calculate the centrality score of each sentence (or vertex). Traditionally, this score is measured by ranking algorithms (Mihalcea and Tarau, 2004; Erkan and Radev, 2004) based on PageRank (Brin and Page, 1998). The sentences with the top score are extracted as a summary. The undirected graph algorithm computes the sentence centrality score as follows:

$$Centrality(s_i) = \sum_{j=1}^{N} e_{ji} \quad (5)$$

After obtaining the centrality score for each sentence, sentences are sorted in reverse order and the top ranked are included in the summary. GUSUM includes the vertex weights of the sentence graph in the calculation of the centrality. Thus, as a first step, the initial rank values of the sentence graph are determined.

The second step to build the sentence graph is to generate the edges that represent semantic sentence similarities. Cosine similarity can be used as a measure to find similarity between sentences of the graph. In this step, all the pairwise Cosine similarities are gathered in a matrix. Cosine similarity is defined as:

$$Cosine\, Similarity = \frac{\sum_{i=1}^{N} A_i B_i}{\sqrt{\sum_{i=1}^{N} \mathrm{A}_i^2}\sqrt{\sum_{i=1}^{N} \mathrm{B}_i^2}} \quad (6)$$

(where A_i and B_i are the components of vector A and B respectively)

Let $D = (s_1; s_2; ...; s_n)$ be a document. We produced using sentence feature scores, $V = (v_1, v_2, ..., v_n)$ is the vertex set where v_i is the representation of sentence s_i. $(e_1; e_2; ...; e_n)$ is a set of vectors , where e_i is the sentence embedding of s_i. Its edges are weighted according to the cosine similarities of the corresponding sentence embeddings. Next, we compute the matrix A with 7:

$$A[i, j] = Cosine\, Similarity(e_i; e_j) \quad (7)$$

Thus, matrix A can be interpreted as the adjacency matrix of an undirected weighted complete graph.

3.4 Ranking and Summary Selection

We propose a variation of weighted undirected graph-based ranking in this section. Based on the idea that the most important sentence in a document is the sentence most similar to all other sentences according to the similarity metric, we modify Equation 5 to include the vertex weights. As a consequence, we define the importance rank for each sentence as follows:

$$Rank(s_i) = v[i] * \sum_{j=1}^{n} A[i, j] \quad (8)$$

where v is the corresponding feature score for s_i, e_i and e_j are the corresponding Sentence-BERT sentence embedding for s_i and s_j .

[1] https://www.sbert.net/

We finally rank and select sentences with Equation 9. The number of sentences in the summary is represented by the k value.

$$summary = topK(\{Rank_{(si)}\})_{i=1,\ldots,n} \quad (9)$$

where the top-ranked k sentences will be extracted as summary.

4 Experimental Setup

In this section we assess the performance of GUSUM on the document summarization task. We first introduce the datasets that we used, then give our pre-processing and implementation details.

4.1 Summarization Datasets

CNN/DM dataset contains 93k articles from CNN, and 220k articles from Daily Mail newspapers, which uses their associated highlights as reference summaries (Hermann et al., 2015). We use the test set which includes 11490 documents provided by hugging face version 3.0.0[2] (See et al., 2017).

NYT dataset contains over 1.8 million articles written and published by the New York Times between January 1, 1987 and June 19, 2007 and summaries are written by library scientists. Different from CNN/DM, salient sentences are distributed evenly in each article. We use The New York Times Annotated Corpus provided by the Linguistic Data Consortium[3] (Sandhaus, 2008). We filter out documents whose summaries are between January 1, 2007 and June 19, 2007 and documents whose length of summaries are shorter than 50 tokens and finally retain 6508 documents (Zheng and Lapata, 2019) .

PubMed & arXiv datasets are two long documents datasets of scientific papers. The datasets are obtained from arXiv and PubMed OpenAccess repositories. The summaries are created from the documents. PubMed contains 215k and arXiv contains 113k documents. We use test sets which includes 6658 documents for PubMed and 6440 documents for arXiv provided by hugging face[4].

Datasets	#docs	avg. doc. length (word)	avg. doc. length (sent.)	avg. sum. length (word)	avg. sum. length (sent.)
CNN/DM	11490	773.22	33.36	57.75	3.79
NYT	6508	1109.10	32.17	96.31	1.18
PubMed	6658	3142.92	101.60	208.02	7.58
arXiv	6440	6446.10	250.36	166.72	6.22

Table 1: Statistic of our CNN/DM , NYT, PubMed and arXiv datasets

4.2 Implementation Details

In GUSUM, during the pre-processing stage, NLTK (Natural Language Toolkit)(Bird and Loper, 2004) was used to collect corpus statistics and process documents using methods such as sentence segmentation, word tokenization, Part of Speech (POS) tagging and using regular expressions to remove parenthesis and some characters.

In the process of creating the graph, we first applied Equations 1, 2, 3 and 4 to calculate sentence feature scores and defined the sums of the obtained values as vertex weights. Next, we calculated the edge weights representing the sentence similarities. For each dataset, we used the publicly released Sentence-BERT model *roberta-base-nli-stsb-mean-tokens* [5] to initialize our sentence embeddings. The *bert-base-nli-mean-tokens*[6] model was also tested in our experiments. However, the *roberta-base-nli-stsb-mean-tokens* showed slightly higher performance (see Table 6). Alternative models that can be applied in our method are listed on Github[7]. In this manner, the model maps sentences and paragraphs to a 768-dimensional dense vector space.

In our experiments, Cosine distance and Euclidean distance were tested to measure the distances between sentence embedding vectors. However, it was observed that higher performance was obtained with the Cosine similarity (see Equation 6) method of Sentence-BERT (see Table 6). The scores obtained as a result of similarity measure were assigned as the edge weight of the graph.

In the last stage, we ranked the sentences using Equation 5 and determined the three most important sentences that should be included in the summary. Table 2 presents a sample golden reference summary and the summary created by GUSUM.

[2] https://huggingface.co/datasets/cnn_dailymail

[3] https://catalog.ldc.upenn.edu/LDC2008T19

[4] https://www.tensorflow.org/datasets/catalog/scientific_papers

[5] https://huggingface.co/sentence-transformers/roberta-base-nli-stsb-mean-tokens

[6] https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

[7] https://github.com/tubagokhan/GUSUM/blob/main/QAforHumanEvaluation.json

Gold-standard Reference
Food and Drug Administration has not found rat poison in pet food that has been killing cats and dogs, but it has found melamine, chemical commonly used to make plastic cutlery that is also used in fertilizer. Mationwide pet food recall , which has involved wet foods all manufactured by Menu Foods and sold under variety of brand names is expanded to include one brand of dry cat food made by Hills Pet Nutrition. brand was found to have been made with batch of wheat gluten shipped to US from China that FDA says was laced with melamine
GUSUM
The Food and Drug Administration said yesterday that it had not found rat poison in pet food that has been killing animals, but that it had found melamine, a chemical commonly used to make plastic cutlery that is also used in fertilizer. Scientists found melamine, which is used as a slow-release fertilizer in Asia, in the urine of cats sickened by the recalled pet foods made by Menu Foods, officials said at a news conference. The recalled pet food has been blamed for at least 16 deaths of pets. Additionally, F. D. A. officials said that they did not believe the contaminated wheat gluten had entered the human food supply, but that they were testing all wheat gluten imported from China for melamine.

Table 2: An example summary generated by GUSUM compared with gold-standard summary

5 Results

5.1 Automated evaluation

ROUGE (Lin and Hovy, 2003) was used to assess the quality of summaries from different models. We report the full length F1 based ROUGE-1, ROUGE-2, ROUGE-L on both CNN/DM, NYT, PubMed and arXiv datasets. The py-rouge package[8] is used to calculate these ROUGE scores.

Table 3 and Table 4 summarize our results on the CNN/DM and NYT short document dataset and arXiv and PubMed long document datasets respectively. The first blocks present the results of strong unsupervised baselines LEAD-3, TEXTRANK (Mihalcea and Tarau, 2004)), LEXRANK (Erkan and Radev, 2004) previous unsupervised graph-based methods. LEAD-3 simply selects the first three sentences as the summary for each document. TEXTRANK (Mihalcea and Tarau, 2004) displays a document as a graph with sentences as nodes and edge weights using sentence similarity and bases PageRank (Brin and Page, 1998) when selecting the best scores. LEXRANK (Erkan and Radev, 2004) also calculates the significance of sentences in representative graphs based on a measure of eigenvector centrality (based on node centrality). The second blocks shows recent supervised methods. For supervised extractive models, we compare with PTR-GEN (See et al., 2017), REFRESH (Narayan et al., 2018a), BertEx (Liu and Lapata, 2019) , Discourse-aware (Cohan et al., 2018), SummaRuNNer (Nallapati et al., 2017) and GlobalLocalCont (Xiao and Carenini, 2019). The third blocks includes recent state-of-the-art unsupervised graph-based methods for document summarization. PACSUM (Zheng and Lapata, 2019), FAR (Liang et al., 2021), STAS (Xu et al., 2020) and Liu et al. (2021) are detailed in Section 2. The last blocks in Table 3 and Table 4 reports results of our method, GUSUM.

As can be seen in Table 3, GUSUM achieves the highest ROUGE F1 score, compared to all other graph-based unsupervised methods on both CNN/DM and NYT datasets. From the results, we can see that our method outperforms all strong baselines in the first block. Furthermore, our method achieves better results than PACSUM and FAR on both datasets. When we compare our method with STAS, our method produces better results, except for the F-1 R-2 metric on CNN/DM. The success of GUSUM can be seen when the latest state-of-the-art unsupervised graph-based method by Liu et al. (2021) and GUSUM is compared. Moreover, it is seen in Table 4, GUSUM also performed very well on arXiv and PubMed long document datasets. Especially F1 R-L provides very high results compared to all other studies.

5.2 Human evaluation

In addition to the Rouge metric, we also evaluated the system output via human judgments. In the experiment, we evaluated the extent to which our approach retained important information in the document, following a question-answer (QA) paradigm used to evaluate the summary quality and text compression (Narayan et al., 2018b).

We created a set of questions based on the assumption that gold-standard summaries highlight the most important content of the document. Then, we examined whether participants could answer these questions simply by reading the system summaries without accessing the article. We created 71 questions from 20 randomly selected documents for the CNN/DM datasets and 59 questions from 18 randomly selected documents for the NYT dataset. We wrote multiple fact-based question-answer

[8] https://pypi.org/project/py-rouge/

Method	CNN/DM			NYT		
	R-1	R-2	R-L	R-1	R-2	R-L
LEAD-3	40.49	17.66	36.75	35.50	17.20	32.00
TEXTRANK (Mihalcea and Tarau, 2004)	33.85	13.61	30.14	33.24	14.74	29.92
LEXRANK (Erkan and Radev, 2004)	34.68	12.82	31.12	30.75	10.49	26.58
PTR-GEN (See et al., 2017)	39.50	17.30	36.40	42.70	**22.10**	38.00
REFRESH (Narayan et al., 2018a)	41.30	18.40	35.70	41.30	22.00	37.80
BertExt (Liu and Lapata, 2019)	43.25	**20.24**	39.63	-	-	-
PACSUM (Zheng and Lapata, 2019)	40.70	17.80	36.90	41.40	21.70	37.50
FAR (Liang et al., 2021)	40.83	17.85	36.91	41.61	21.88	37.59
STAS (Xu et al., 2020)	40.90	18.02	37.21	41.46	21.80	37.57
Liu et al. (Liu et al., 2021)	41.60	18.50	37.80	42.20	21.80	**38.20**
GUSUM	**43.40**	17.02	**42.38**	**43.64**	22.01	37.90

Table 3: Test set results on the CNN/DM and NYT datasets using ROUGE F1. Results are taken from (Liang et al., 2021)

Method	arXiv			PubMed		
	R-1	R-2	R-L	R-1	R-2	R-L
LEAD-3	33.66	8.94	22.19	35.63	12.28	25.17
TEXTRANK (Mihalcea and Tarau, 2004)	24.38	10.57	22.18	38.66	15.87	34.53
LEXRANK (Erkan and Radev, 2004)	33.85	10.73	28.99	39.19	13.89	34.59
PTR-GEN (See et al., 2017)	32.06	9.04	25.16	35.86	10.22	29.69
Discourse-aware (Cohan et al., 2018)	35.80	11.05	31.80	38.93	15.37	35.21
SummaRuNNer (Nallapati et al., 2017)	42.81	16.52	28.23	43.89	18.78	30.36
GlobalLocalCont (Xiao and Carenini, 2019)	**43.62**	**17.36**	29.14	44.85	**19.70**	31.43
PACSUM (Zheng and Lapata, 2019)	39.33	12.19	34.18	39.79	14.00	36.09
FAR (Liang et al., 2021)	40.92	13.75	35.56	41.98	15.66	37.58
GUSUM	40.98	11.76	**39.49**	**44.98**	16.26	**43.98**

Table 4: Test set results on the arXiv and PubMed datasets using ROUGE F1.Results are taken from (Liang et al., 2021)

Method	CNN/DM		NYT	
	Score	%	Score	%
LEAD-3	54.75	77.11	42.00	71.19
TEXTRANK	56.38	79.40	39.50	66.95
GUSUM	**57.00**	**80.28**	**46.25**	**78.39**

Table 5: Results of QA-based evaluation on CNN/DM, NYT. We compute a system's final score as the average of all question scores.

pairs for each gold summary. Our Question and Answer set is available at https://github.com/tubagokhan/GUSUM/blob/main/QAforHumanEvaluation.json.

We compared GUSUM against LEAD-3 and TEXTRANK on CNN/DM and NYT. We used the same scoring mechanism from Ziheng and Lapata (2019), a correct answer was marked with a score of one, partially correct answers with a score of 0.5, and zero otherwise. The final score for a system is the average of all its question scores. Four fluent English speakers answered the questions for each summary. The participants were chosen from university volunteers who gave their consent to contribute to the study.

The results of our QA evaluation are shown in Table 5. Based on summaries generated by LEAD-3 participants can answer 77.11% and 71.19% respectively CNN/DM and NYT of questions correctly. Summaries produced by TEXTRANK have 79.40% and 66.95% scores. When the scores of GUSUM are compared with the scores of the other two systems, the high performance of GUSUM is seen. The main reason for GUSUM's slightly higher performance in CNN/DM dataset compared to NYT is thought to be the use of human-generated gold summaries in NYT. Another possibility is that the summaries created from the CNN/DM dataset are shorter and users can focus more. It is thought that the participants have a leaning to become distracted with the longer summaries in the NYT dataset compared to CNN/DM.

5.3 Ablation Study

In order to access the contribution of three components of GUSUM, we remove or change each component of them and report ablation study results in Table 6. Since short and long documents have different structures, separate experiments are carried out. In Table 6, the results of the NYT dataset in the first block and the PubMed dataset in the second block are presented.

NYT			
	R-1	R-2	R-L
GUSUM	**43.64**	**22.01**	**37.90**
-Removed All Sentence Features	36.63	14.91	30.58
-bert-base-nli-mean-tokens	43.28	21.73	37.48
-Eucludian Distance	35.35	16.43	31.10
PubMed			
GUSUM	**44.98**	**16.26**	**43.98**
-Removed All Sentence Features	44.08	15.53	43.32
-bert-base-nli-mean-tokens	44.27	15.66	43.36
-Eucludian Distance	37.77	11.29	37.40

Table 6: Ablation study results on NYT and PubMed datasets using ROUGE F1.

We can observe that sentence feature scoring is critical to GUSUM's performance, mainly on NYT. When all sentence features are eliminated, the performance of GUSUM drops sharply. In another experiment, we replaced the *roberta-base-nli-stsb-mean-tokens* model with the *bert-base-nli-mean-tokens* model in both datasets and discovered just a minor difference in performance. In our last experiment, we changed the method of measuring the similarity of sentence embeddings to generate the graph. When we employ the Euclidean method, there is a dramatic decrease in the performance of GUSUM.

6 Discussion

There are two basic stages in document summarization: (1) Identification of the most salient sentences in the document, (2) Removal of similar sentences from the summary. Generally in graph-based approaches, graphs are created based on only sentence similarity, and then the most salient sentences are selected. On the contrary, in GUSUM we included these two basic steps in our approach. Along with the semantic similarity, we also embedded the attributes of the sentences in our graph. Furthermore, GUSUM advocates the idea that the most important sentence in a document is the sentence most similar to the others. For this reason, the total similarity value for each sentence is evaluated in the ranking stage. The experimental results of GUSUM, which is a simple and effective method based on these ideas, prove the validity of our ideas.

As seen in the experimental results, GUSUM showed high performance on all datasets. However, the limitation of GUSUM is that sentence features scoring does not have a significant impact on long documents as can be see in Table 6. The main reason for this situation is that the ranking algorithm we use in long documents produces results that are very close to each other. Therefore, we argue that for long documents, sentence feature scores should be enriched by including thematic word, sentence centrality, title similarity, the similarity to the first sentence, cue-phrases, term weight scores, etc. Moreover, adding section segmentation for long document summarization can significantly improve performance.

The most difficult part of this study is the evaluation stage. Evaluating the performance of summarization systems poses a problem for many researchers (Schluter, 2017). It is a known fact by researchers that human evaluation is the best summary performance evaluation method. For this reason, we included human evaluation as a performance evaluation method in our study. However, what we noticed in our study is that the questions used for human evaluation based on the QA paradigm in other studies published to date have not been shared by the researchers. As a result of this situation, researchers prepare their own questions and the results cannot be compared with the literature. As a solution to this problem, we publish the questions and answers that we prepared from the CNN/DM and NYT datasets based on the QA paradigm for use in future studies (See 5.2).

7 Conclusions and Future Works

In this paper, we have proposed a graph-based single-document unsupervised extractive summarization method. We revisited traditional graph-based ranking algorithms and refined how sentence centrality is computed. We defined values indicating the importance of the sentences in the document for the node weights in the graphs and we built graphs with undirected edges by employing Sentence-BERT to better capture sentence similarity. Experimental results on four summarization benchmark datasets demonstrated that our method outperforms other recently proposed extractive graph-based unsupervised methods and achieves performance comparable to many state-of-the-art supervised approaches which shows the effectiveness of our method.

In the future, we would like to remove the limitations that would increase the performance of GUSUM in long document summarization with the ideas introduced in this study and explore the performance of GUSUM in multi-document summarization.

References

Sanghwan Bae, Taeuk Kim, Jihoon Kim, and Sanggoo Lee. 2019. Summary level training of sentence rewriting for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 10–20, Hong Kong, China. Association for Computational Linguistics.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117. Proceedings of the Seventh International World Wide Web Conference.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal Of Artificial Intelligence Research*, 22(1):457–479.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Improving unsupervised extractive summarization with facet-aware modeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1685–1697, Online. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 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*, pages 150–157.

Jingzhou Liu, Dominic J. D. Hughes, and Yiming Yang. 2021. *Unsupervised Extractive Text Summarization with Distance-Augmented Sentence Graphs*, page 2313–2317. Association for Computing Machinery, New York, NY, USA.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3075–3081. AAAI Press.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, Volume 1 (Long Papers), pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Ani Nenkova, Sameer Maskey, and Yang Liu. 2011. Automatic summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, page 3, Portland, Oregon. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Evan Sandhaus. 2008. The New York times annotated corpus ldc2008t19. *Linguistic Data Consortium, Philadelphia*.

Natalie Schluter. 2017. The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Nikhil S. Shirwandkar and Samidha Kulkarni. 2018. Extractive text summarization using deep learning. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–5.

Ladda Suanmali, Mohammed Salem Binwahlan, and Naomie Salim. 2009. Sentence features fusion for text summarization using fuzzy logic. In *2009 Ninth International Conference on Hybrid Intelligent Systems*, volume 1, pages 142–146.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Shusheng Xu, Xingxing Zhang, Yi Wu, Furu Wei, and Ming Zhou. 2020. Unsupervised extractive summarization by pre-training hierarchical transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1784–1795, Online. Association for Computational Linguistics.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy. Association for Computational Linguistics.