International Journal of Computer Vision and Signal Processing, 1(1), 27-38(2020) ORIGINAL ARTICLE

Hybrid Text Summarizer for Bangla Document

Mahimul Islam*, Fariha Nuzhat Majumdar, Asadullahhil Galib, Md Moinul Hoque

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology, Bangladesh

Abstract

Automatic text summarization is needed to concisely extract a small subset of text portions from a large text where the isolated text may have sentences that are more significant compared to other sentences in the text. Although there have been a lot of approaches to English text summarization, very few works have been done on automatic Bengali text summarization. For the evaluation purpose, a dataset was formulated from the scratch with Bengali news documents from two reputed newspapers. The evaluation dataset was classified into four different classes with benchmark standard summary text, generated by a group of random human contributors for each of the documents. The current work presents a hybrid approach for dealing with the summarization process of Bengali text documents. The hybrid model is introduced with a goal to improve the overall accuracy of the summary text generation. The proposed model generates a summary text based on keyword scoring, sentiment analysis, and the interconnection of sentences. After conducting the evaluation on the existing dataset, the proposed system performs with an average of 0.77 Recall Score, 0.57 Precision Score, and 0.64 F-measure Score. Empirical verification with other similar systems shows that the proposed model can be used as an alternative system to address the Text Summarization problem of Bengali documents.

Keywords: Hybrid Bangla Document Summarization, Sentence Scoring, Sentiment Analysis, Keyword Ranking, Text Ranking

ISSN: 2186-1390 (Online)
http://cennser.org/IJCVSP

Article History:
Received: 28 Feb 2020
Revised: 11 June 2020
Accepted: 23 Nov 2020
Published Online: 23 Nov 2020

1. INTRODUCTION

Text summarization requires a short, accurate, and fluent summary of a longer text document. From the summary, important information can be gained, making the overall procedure more comfortable, and fewer resources are needed. To discover relevant information faster from a huge number of text documents available online, automatic text summarization ideas have been found very significant. A few methods have been explored for the generation of summary from Bengali documents. If the summary contains sentences from the document's major topics, it has a better chance of giving a better perspective of the document. The summary generation approach of the proposed system is extractive, i.e., they contain sentences as it appears in the document.

According to Sarkar [1], text summarization involves preprocessing, stemming, sentence ranking, and summary generation. The preprocessing step requires removal of stopwords, stemming and converting the input into a collection of sentences. Uddin and Khan [2] described an extraction-based method for summarizing Bengali documents. Different features, such as location, term frequency, numerical data, etc., were used to rank the sentences. Based on the features, they have designed the Bengali summarizer and concluded that the summary size should be 40 percent of the actual content. Das and Bandyopadhyay [3] have summarized Bengali documents using sentiment information. They have tried to identify the sentiment information in a document and then aggregated that for generating the summary. Mihalcea [4] [5] has focused on text summarization based on graphs. A graph can be constructed considering the sentences as nodes and connecting them with edges. After that, edge weights may be measured by calculating the similarity between two nodes.

Keeping the state-of-the-art in view, a hybrid Bangla Text Summarizer has been proposed in this work, which combines the following methods:

- Sentiment Scoring
- Keyword Ranking
- Text Ranking

In the proposed method, the top 40 percent of the actual document was considered as a generated summary based on the

*Corresponding author
Email addresses: mahimulislam@gmail.com (Mahimul Islam), nivamila@gmail.com (Fariha Nuzhat Majumdar), agalib.aust@gmail.com (Asadullahhil Galib), moinul@aust.edu (Md Moinul Hoque)

combined weighted score, which we describe in an upcoming section.

People may often overlook important phrases, but computers cannot skip them, and thus important phrases will always be listed. With the digitization of media along with publishing, this technique saves their reading time for those who have no time to go through the whole post, document, or book, as they don't have to read massive quantities of pointless and redundant data. While notable works have been done for English and other languages, the Bengali text summarization has often been ignored, despite being spoken by a significant number of people. On the internet, there are millions of Bengali papers, which are massive in size and need to be reduced to make them more readable.

Bengali text summarization is a very difficult task, mainly because of the small number of openly accessible resources. We generated our very own dataset from scratch due to the lack of an openly accessible dataset. About 520 news documents were collected from two famous Bengali newspapers, and we used about half of them to train our model. Then, we combined three popular summary generation techniques to create our hybrid model. Evaluating a summary is a very difficult task since there is no flawless summary. Two scholars can generate two summaries from a single text, and the summaries they produce cannot be the same. Three individual model summaries produced by two groups of scholars were compared with the summaries developed by our system for the purpose of evaluation. Three different scores of evaluation measures were determined for each of the summaries produced, and the average score was considered. We also measured the time it takes to summarize the documents.

The rest of the paper contains: Related works in section 2. The detailed proposed model in section 3. Evaluation Measures, Experimental Results and Comparison with other methods in section 4. Finally, the conclusion and future works in section 5.

2. RELATED WORKS

In this section, an overview of several types of research relevant to automatic text summarization has been discussed. Most of the research work on text summarization is based on English documents. Despite Bengali being the 7th language according to the number of speakers in the world, very few researches were conducted on automatic Bengali text summarization.

Sarkar [1] [6] has discussed text summarization for the single document of Bengali language signifying the impact of thematic term feature and position feature of sentences. In linguistics, thematic feature means to relate to the theme of writing. The work had mainly three phases: preprocessing, sentence ranking, and summary generation. Sentence ranking has been done with thematic terms and sentence position. The average unigram-based Recall score is 0.4122 and the score for their baseline is 0.3991. In his other research [7], he has presented a key-phrase based approach for summarization, which focused on extracting a set of key phrases from a document and generating an extractive summary based on that. Key phrases can be single or multi-word. He has used two different datasets, one for English and another for Bengali. He concluded that the results were quite satisfactory in comparison with the previous works. Srivastava and Gupta [8] have attempted an Extract Technology based approach that emphasizes summary generation based on the frequency of words in their research. They have proposed a technique based on NLP (Natural Language Processing), which is known as the Gradual NLP algorithm. The summing process can be broken down into three stages: analysis, development, and synthesis. The analysis phase analyzes the text of the data and selects several key characteristics. The transformation process turns the empirical findings into a summary representation. After that, the synthesis process takes the summary representation and produces a suitable summary that corresponds to the user needs. The algorithm counts the total frequency of words other than stopwords and then calculates the average frequency. For summary generation frequency of the sentences with cue words present in them are increased and selected for a summary if the score is greater than the average frequency. Chandro et al. [9] have experimented with extraction-based summarization techniques by collaborating individual words and scoring sentences. Experimentation documents were collected from the popular Bengali daily newspapers. They have done sentence ranking based on Term Frequency, Positional Value, Connecting Words, and Sentence length of the document. Combining these parameters, sentences were ranked, and K-top ranked sentences were picked for the summary. The average unigram-based Precision, Recall, and F-measure scores were 0.80, 0.67, and 0.72, respectively. Uddin and Khan [2] experimented with Bengali text summarization and they have put significance on sentence location, cue phrase presence, title word presence, term frequency, and numerical data. They have argued that sentences that appear in the first or last of passage are of more importance. Moreover, the presence of cue phrases, words from titles, words with high frequency, and numerical data also put importance on a sentence. They have achieved an average accuracy of 71.3 percent. Efat et al. [10] have discussed Bengali summarization taking into consideration several attributes. They have calculated a sentence's scores based on frequency, sentence position, cue phrases, etc. After calculating scores based on various aspects, the final sentence scores have been calculated as a weighted summation of the scores of individual features. They have presented that 83.57 percent of summary sentences match to human-generated summaries. Haque et al. [11] have discussed Bangla summarization using key phrases. They have sorted sentences in ascending order based on their scores, and sentences with numerical figures have been given importance. After combining the scores, sentences have been ranked. Dataset has been made with four hundred newspaper documents that are of wide varieties. Using ROUGE-1 and ROUGE-2, they said that the quality of their summaries has improved.

Das and Bandyopadhyay [3] have summarized Bengali documents using sentiment information. They have used a classifier based on the support vector machine. Three kinds of features have been considered, which are lexico-syntactic, syntactic, and discourse level. Parts of speech, SentiWordNet, frequency, stemming, chunk label, dependency parsing depth, the title of a document, first paragraph, term distribution, and collocation have been used as features in the work. It has been said that the summarization system has achieved a Precision of 72.15 percent, Recall of 67.32 percent, and F-Measure of 69.65 percent. Mandal et al. [12] have used the Particle Swarm Optimization (PSO) method for sentiment analysis in their research of text summarization. The suggested methodology follows the basic concepts of PSO and replaces fitness function with sentiment ranking. The work mainly involves pre-processing, evaluation of fitness value that is used by the PSO by sentiment score, and generation of two sets of summary (namely summary by clustering based PSO and GA and constraints based PSO). In pre-processing, sentences are grouped after removing the special characters and stopwords. For fitness value evaluation, sentiment score (measured bySentiWordNet) is used to evaluate the best particle. Then cluster number is determined, and an automatic population partitioning (APP) for sentence clustering is implemented. The PSO was applied in the APP model, and the fitness value is determined by the similarity of the cluster. For evaluation, five different datasets from several websites have been collected and the optimized summaries were evaluated. ROUGE-1 and ROUGE-2 scores have been considered for evaluating the summaries. The average ROUGE-1 Precision, Recall, and F-measure scores of the system are 0.4352, 0.4465, and 0.4399, respectively, and the average ROUGE-2 Precision, Recall, and F-measure scores of the system are 0.2154, 0.1852, and 0.1990, respectively. Roul and Sahoo [13] have proposed a work that focuses on summing up feedback for films bought from Amazon using a combination of four state-of-the-art algorithms and a search technique for features. Sentiment analysis was carried out to categorize the reviews into positive and negative. In addition, a novel approach called hierarchical summarization is attempted to summarize broad reviews into a summary of a few sentences. To decide the optimal summary, the results of the machine-generated summaries are compared with the existing algorithms using the ROUGE score. They proclaimed that the proposed system has shown promising outcomes. For the purpose of summing up, Yadav and Chatterjee [14] proposed a computationally efficient technique based on the sentiment of keywords in the text. The proposed methodology includes sentiment computation of sentences that can be used to determine the relevant and most significant sentences of a document. For this research, they developed and tested three models S1, S2, and S3, where S1 is the total sentiment, S2 is the absolute sentiment, and S3 is the average sentiment of the sentence. They tested the results on the standard DUC2002 dataset and compared them with different summarization approaches, Random indexing based, LSA based, Graph-based, and Weighted graph-based methods for different percentages of summarization. They claimed that the suggested scheme had been found efficient for 50 percent summarization in particular.

Haque et al. in their other work [15] have done text summarization with Bengali documents using sentence ranking and clustering. Sentences have been ranked with term frequency calculation for each sentence and sentence frequency. If an overlap ratio of two sentences has been shown over or equal to sixty percent, then the smaller sentence falls out of consideration, and the importance of larger sentences increases. Sentences have been clustered using cosine similarity to group similar sentences. Then the summary has been generated by selecting sentences from clusters based on the volume of clusters. After evaluation, Precision, Recall, and F-score values have been calculated as 0.608, 0.664, and 0.632, respectively. Li et al. [16] have used a keyword extraction method based on TextRank that uses the essentiality ranking of words in documents. They attempted to use Word2Vec and Doc2Vec to improve keyword extraction of short text. Word2Vec was used for training word vectors to obtain the semantic information between words. Doc2Vec was used for training the paragraph vectors and to increase the accuracy of keyword extraction through coordinated word vectors and paragraph vectors. The candidate keyword graph was constructed to represent the structural relationships between the sentences, and Word2Vec and Doc2Vec were used to capture the semantic details between words. They added the collaborative training method for word vectors and paragraph vectors first and then used the clustering nodes of the TextRank model. The weights of the keywords that were generated by computing the jump probability between nodes were adjusted, and then the node-weighted score was obtained, and eventually, the generated keywords were sorted. For evaluation, they checked Southern Weekend News' long document dataset, and Sina Weibo's short text dataset. The F-measure reaches a limit of 43.1 percent when the number of extracted keywords is 7. The experimental results suggest that the improved approach works well on the dataset. Hu et al. [17] have described that their research method is divided into five key steps: hotel review collection, review pre-processing, sentence importance calculation, sentence similarity calculation, and top-k sentence recommendations. Two sets of reviews for the two hotels posted on TripAdvisor.com were gathered to evaluate the efficiency of the proposed system. The pre-processing activities included tagging of part-of-speech (POS), deleting stop-words, filtering POS, and selecting sentences. In this analysis, several factors that affect a review's sentencing value are considered: the responsiveness of a review author, the helpfulness of a review, the time required for reviewing, and the content of the sentences in a review. Two forms of similarity, content and similarity of sentiment were considered, which used nouns and adjectives, respectively in similarity calculations. This approach exhibited scores of 2.8 and 2.75 in k = 5 and k = 10 and 2.5 and 2.75 in k = 5 and k = 10 respectively for the Red Roof Inn and the Gansevoort Meatpacking hotel. Basheer et al. [18] have modified the weighted TF IDF algorithm to summarize books into specific keywords. They compared the changed algorithm with the existing TextRank Algorithm, Luhn's Algorithm, LexRank Algorithm, and Latent Semantic Analysis(LSA). From the comparative analysis, Weighted TF IDF is found to be an efficient algorithm for automating text summarization and generating an efficient summary, and translating from text to speech. The software is divided into three main functions: pre-processing, selection of functionalities, and description. In preprocessing, text-specific NLTK functions such as tokenization, trailing, POS tagger, and stopwords have been performed. After calculating each word's TF-IDF value, the information can be used to determine a sentence's value. 3–5 sentences with the full sense of TF-IDF are selected. In their work, they showed a comparison between

TextRank and LexRank. TextRank is derived from the PageRank algorithm where the sentences are considered as graph vertices, and the edge weights between sentences measure the degree of similarity of two sentences. LexRank, though similar to TextRank, is unsupervised. LexRank uses Cosine as the attribute for calculating the similarity between two sentences. LexRank measures the distance in the middle of two sentences. The cosine angle is dependent on the relevancy between the two. For evaluation purposes, ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE-p, and ROUGE-w of TextRank, Luhn's, Lex Rank, LSA, and Weighted TF-IDF were calculated. Maximum F-measure values were achieved for weighted TF-IDF. ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE-p, and ROUGE-w F-scores of weighted TF-IDF are 89.36, 88364, 87.89, 87.12, 90.92, 48.68 respectively. Xiong et al. [19] have proposed a method of speech text analysis where a heuristic algorithm for clustering speech texts and obtaining similar text sets is used. An enhanced TextRank algorithm is then used to generate multi-document summaries, and the summary results are fed back to the experts. The multi-document summarization approach is based on TextRank, which quantifies sentence location in paragraphs, key sentence weight, and sentence length. The work is divided into three parts: text pre-processing, text clustering to get similar speech text sets, and multi-document summarization. Pre-processing involves word segmentation, stop-words deletion, and text feature extraction. For feature selection, TF-IDF has been used, and then, the documents have been vectored using the Vector Space Model. The cosine of the angle between two vectors is used for the similarity measure of two texts. The multi-document summaries have been generated using TextRank and improved the TextRank algorithm where they considered the position of sentences in a paragraph, key sentence processing, and sentence length filtering. Eventually, a prototype is developed to check the validity of the method using the four parameters of recall rate, accuracy rate, F-measure, and user assessment. For evaluation of the effectiveness of the method, they compared the summary results of this article with those produced by TextRank. For a summary ratio of 20 percent, the F-measure score of the improved method is 0.698, and for 30 percent, this score is 0.648. They proclaimed that the experimental results indicate a strong performance of the process in the system.

Uçkan and Karcı [20], Mutlu et al. [21], and Joshi et al. [22] worked on extractive text summarization methods. All of them used the DUC 2002 dataset in their respective works. Uçkan and Karcı [20] proposed a method based on graph independent sets. The method achieved 0.38072, 0.51954, and 0.59208 ROUGE Recall score for 100, 200, and 400-word summaries, respectively. The ROUGE F-Measure score for 200-word summaries were 0.4973. Mutlu et al. [21] used fuzzy systems dependent on a feature vector and a fuzzy rule set for the summary generation. They reported an F-Measure score of 0.6587. Joshi et al. [22] developed a deep auto-encoders-based system for summary text generation. They used an unsupervised framework. This system achieved an F-Measure score of 0.5170.

From the state-of-the-art, it was clearly visible that there were scopes of improvement in ROUGE scores. Most of the works were mainly based on the English language. None of the previous works except Das and Bandyopadhyay [3], has labeled the data using any classifier. The works were mainly focused on either Keyword [1-2] [6-11] , Sentiment analysis [3] [12-14] , or Sentence inter-connectivity (TextRank, LexRank, LSA) [15-19]. The problem of only considering the keywords is that, despite working very well for classifying documents, it does not consider the relationship between two sentences, which is very important for summarizing. On the other hand, considering only the interconnected sentences fail to classify the document, and many crucial words containing sentences get missing. Summaries should be neutral. Keeping that in mind, Sentiment Analysis should be done while generating summaries. Even doing so simply would skip the major stuff the other two strategies achieve independently. Apart from that, none of the papers discussed the time generated to create a summary. Therefore, we decided to combine all of the three methods and analyze them to get the optimum output. We also carried out the time analysis in order to determine the time needed to produce a single summary.

3. PROPOSED METHOD

In this work, a text summarizer was developed that generates an extractive summary of Bengali documents. Different scores were given to the sentences based on some criteria to select the best ones representing the gist of a given document. The process flow of the proposed model is shown in Figure 1.

The proposed model contains the following steps:

3.1. Input Document

Any Bengali news document can be used as input of the summarizer. After researching different domains, it was found that usually, people are more interested in reading accident, entertainment, economics, and politics related news. So, the top four categories: Accident, Entertainment, Economics, and Politics, were selected to conduct the experiments. The corpus was generated by directly extracting news from the Daily Prothom Alo and the Daily Kaler Kantho newspaper without any modification. About 520 news documents from the mentioned categories were collected. The dataset and news documents are available online [23].

3.2. Preprocessing

Preprocessing involves any type of processing performed on raw data. It was done by splitting the input documents into sentences. Then words were tokenized, and after that stop-words removal technique was used to remove irrelevant words that are frequently used but don't contribute to generating the summary such as অতএব, অথচ, অথবা, etc. For this purpose, a list containing the stop words were used. The list of stop-words is available online [24]. Two different setups were created where one setup used 50 percent of total document for testing purpose and 50 percent of total document for training purpose and the other setup

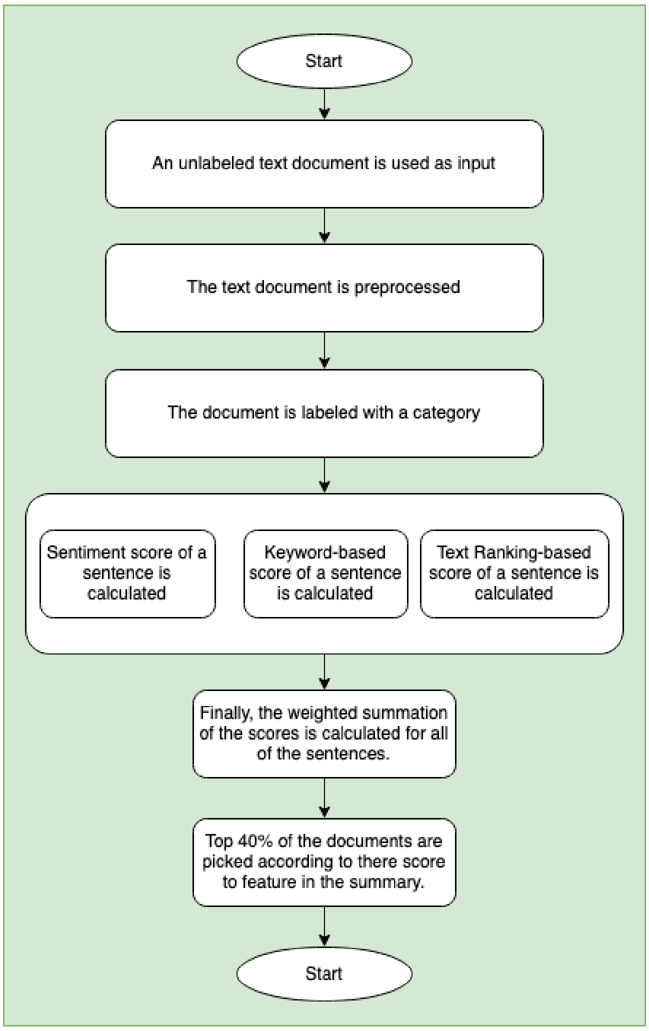
Start
An unlabeled text document is used as input
The text document is preprocessed
The document is labeled with a category
Sentiment score of a sentence is calculated
Keyword-based score of a sentence is calculated
Text Ranking-based score of a sentence is calculated
Finally, the weighted summation of the scores is calculated for all of the sentences.
Top 40% of the documents are picked according to there score to feature in the summary.
Start

Figure 1: Flow chart of the proposed model for Hybrid summarizer for Bengali Document.

used 70 and 30 percent of the documents, respectively. For each setup from each category, a total of eight different lists were created containing the most frequently used words. The score of each word was calculated by using the following formula:

$$S = \frac{F}{N} \tag{1}$$

where, in 1,
S = Score of a specific word
F = Total appearance of a word in all documents
N = Total number of words present in all documents

Sample scores for a few words are, 'সড়ক', '0.0044033822', 'গত', '0.0028365346', 'দুর্ঘটনায়', '0.0028365346', 'নিহত', '0.002647,4323', etc.

3.3. Document Type Separation

Document type separation involves separating documents in different categories. Before summarizing, the input document has to be categorized as a specific type. Classification is needed to group similar kinds of news documents. A word can have different weights in different categories. But in a specific category, if a word is more frequently used, it would get more score. As mentioned earlier, four categories of news: such as Accident, Economics, Entertainment, and Politics, were collected. So, the input document is categorized in any of the stated document types. To do that, a specific list with words and their frequency scores for each category was prepared. For classifying an unknown document, every word of the document was cross-checked with the lists of all four categories. If a match was found, the corresponding score in that category (the same word can have different scores in a different category) was summed up. The class with the highest score among those four was selected as the class of the unknown document.

3.4. Individual Sentence Scoring

Three approaches were combined in a hybrid form to determine the score of each sentence of the given unknown document.

3.4.1. Sentiment Scoring

Machine-generated summaries are free from bias. So, neutral sentences from a document should be picked for being in summary. Chen and Skiena [25] have built sentiment lexicons for about 136 major languages by using graph propagation techniques. A semantic knowledge graph was constructed for propagating lexicons. For each edge between related words, a 5-bit integer was used to store five possible unidirectional semantic links. In total, 7,741,544 high-frequency words were selected from 136 languages as vertices. For the Bengali language, 2393 lexicons were used with a positive and negative ratio of 0.42. For calculating the sentiment polarity of words, only antonym links were considered negative. An edge doesn't get any weight if it has both negative and positive links. Using this model, unique word's polarity can be calculated (+1 for positive words, -1 for negative words, and 0 for neutral words). A sentence can have a score based on the neutral words (the words with polarity score 0) present in that sentence. So, in a sentence, more neutral words mean more sentiment score. For example, 'বারবার', 0, 'কেন', 0, 'ডুবছে', 0, 'নৌযান', 0, '?', 0, etc.

3.4.2. Keyword Ranking

Basheer et al. [18] have proposed a method where the weighted TF-IDF was used for the summary text generation. TF-IDF can be calculated by using the following formulas:

$$TF = \frac{F}{N} \tag{2}$$

where, in 2,
TF = Term Frequency
F = Total appearance of a word in a document
N = Total number of words present in a document

$$IDF = log\frac{N}{D} \tag{3}$$

where, in 3,
IDF = Inverse Document Frequency
N = Total number of documents
D = Document frequency

$$TF - IDF = \frac{TF}{IDF} \tag{4}$$

We have analyzed the dataset, and it was determined that key phrases don't put extra value on top of keywords. Because there is a minimal number of key phrases in the dataset. As already mentioned, there were four categories of documents, and a list for each category with words and respective frequency scores were prepared (in descending order). E.g., 'সড়ক', 'গত', 'দুর্ঘটনায়', 'নিহত', etc., and their respective frequencies are 163, 105, 105, 98, etc. Each sentence can have a score based on each word present in it. The words of each sentence are compared with the list, and scores were calculated. The summation of these scores determines the score of that individual sentence.

3.4.3. Text Ranking

It is a similarity-based ranking model for text processing which, can be used in order to find the most relevant sentences in the text.

Text Ranking requires the tokenization of each sentence from the training dataset and converting them into sentence vectors. To achieve this, a vector model was created with the words from each category. Figure 2 represents a part of the Word2Vec model.

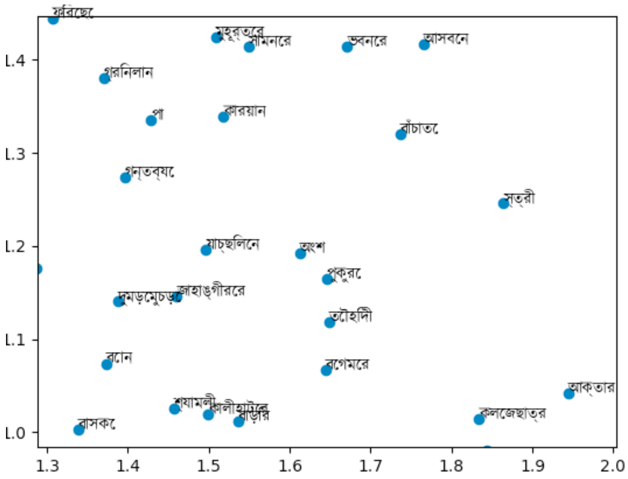

Figure 2: Visual representation of the word to vector model.

After the generation of the model, it was loaded, and using the model, each sentence was represented as a vector. Then each vector was compared with all the other vectors present in the text, and the similarity score with each of them was calculated. The summation of scores defines that specific sentence's similarity score. There are several ways to calculate the similarity score:

Basheer et al. [18] considered the edge weights between sentences for measuring the degree of similarity of two sentences. They have discussed LexRank, which uses Cosine similarity between two sentences. It actually measures the distance between two sentences. The cosine angle is dependent on the relevancy between the two sentences. In most of the cases of their experiment, LexRank outperformed TextRank.

Xiong et al. [19] implemented TextRank based summarization in their paper, and for text similarity calculation, he compared the deviation of angles between text vectors. They have calculated the Cosine of the angle between the vectors.

Deshpande and Lobo [26] proposed a solution for multi-document summarization by using a clustering-based approach and utilized the “cosine similarity measure." After conducting the experiments, they concluded that the method outperformed other similar methods, and clustering redundancy was reduced.

For conducting our experiment, the cosine similarity measure was used for finding relevant sentences. The similarity score between two sentences was calculated using the following formula:

$$S = \frac{V1 \cdot V2}{\|V1\| * \|V2\|} \tag{5}$$

where, in 5,
S = Similarity score of a sentence
V_1 = Vector representation of sentence 1
V_2 = Vector representation of sentence 2

3.5. Candidate Sentence selection and output generation

The three approaches used may calculate different scores for a similar sentence because of variation in their scoring model. To select the highest-ranked sentences suggested by three different approaches, weights were multiplied to each of the sentences. The weights for different approaches were selected empirically. After the multiplication with weight value, the top 40 percent of the sentences (non-overlapped) were selected for the final generated summary. The summarizer has gone through some repeated validation tests and evaluations by setting different weight values to the hybrid model, which combines keyword, sentiment scoring, and text ranking model. Finally, the weights of 0.2, 0.3, and 0.5 were selected for the sentiment scoring method, keyword scoring method, and text ranking based scoring method, respectively.

The scoring of each sentence was done by using the following formula:

$$SentenceScore = SS * 0.2 + KR * 0.3 + TR * 0.5 \tag{6}$$

where, in 6,
SS = Total Sentiment based score of that sentence
KR = Total Keyword-based score of that sentence
TR = Total Text Ranking based score of that sentence

All the sentences of an unknown text document were sorted in descending order, based on their scores. The top 40 percent of total sentences were selected to appear in summary. Sentences were presented in the same order they appear in the original text to be listed in summary.

4. EVALUATION MEASURES AND EXPERIMENTAL VERIFICATIONS

For evaluation purposes, two different datasets were used. In the first phase, a corpus was created from the Daily Prothom Alo and the Daily Kaler Kantho by extracting 520 online news documents from four different categories of news. Two different setups were used. In the first one, 260 news documents were used for testing, and 260 news documents were used for training. In the second one, 364 news documents were used for testing, and 156 news documents were used for training. The summaries to be compared with the system generated summary are considered as Benchmark summaries and they were generated by random human contributors. The second dataset was collected from the Bangla Natural Language Processing Community [27]. This dataset consists of two different setups with 100 documents in each. And three model summaries were collected from two groups of scholars to evaluate our proposed system's generated summary. System generated summaries were evaluated with each of the models, and the average scores were reported.

4.1. Evaluation of the first dataset

4.1.1. Classification Result

Table 1 and 2 show the classification results (confusion matrix) of the proposed system for both setups.

Table 1: Classification Result in 1st setup

Actual Class	**Predicted Class**			
	Acc.	***Eco.***	***Ent.***	***Pol.***
Accident	64	1	0	0
Economics	1	60	1	3
Entertainment	5	13	43	4
Politics	6	14	0	45

Table 2: Classification Result in 2nd setup

Actual Class	**Predicted Class**			
	Acc.	***Eco.***	***Ent.***	***Pol.***
Accident	79	9	3	0
Economics	2	86	1	2
Entertainment	17	21	50	3
Politics	8	14	1	68

From Table 1, it can be seen that, for the first setup the overall accuracy of the proposed classifier is (64 + 60 + 43 + 45) / (520 * 0.5) = 0.815 or 81.5 percent. Accident classification achieved the highest classification prediction score, while entertainment classification got the lowest prediction score. Accident news can be easily identified using certain keywords in the text, such as 'সড়ক', 'দুর্ঘটনায়', 'নিহত', etc. And most of the news documents contain them.

While examining the reason behind the poor score of entertainment class, it was found that the news documents used in the datasets for entertainment class were generally interviews and short stories. The top few keywords from this class were, 'ছবি', 'কাজ', 'অভিনয়', etc. These keywords weren't common in most of the documents. So the accuracy got affected by this.

From Table 2, it can be seen that, for the second setup the overall accuracy of the proposed classifier is (79 + 86 + 50 + 68) / (520 * 0.7) = 0.777 or 77.7 percent. In this case, the Economics class outperformed the Accident class by a minimal margin. The Economics class also has class specified keywords, such as: 'টাকা', 'ঋণ', 'ব্যাংক', etc. Because most of those documents from Economics category used for testing had these words in common, they performed better.

4.2. Evaluation of the generated Summary

For the evaluation purpose ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric was used which is widely used for evaluating the quality of text summarization.

ROUGE-1: It determines the overlap of 1-gram (each word) between the system generated and reference summaries [28].

ROUGE-2: It determines the overlap of bi-grams between the system generated and reference summaries [28]. ROUGE has three main scoring systems, they are Recall, Precision, and F-Measure. They can be calculated using the following formulas:

$$Recall = \frac{O}{G} \tag{7}$$

$$Precision = \frac{O}{R} \tag{8}$$

where, in 7 and 8,
O = Number of overlapping words
G = Number of words in the gold summary
R = Number of words in the reference summary

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

Few empirical experiments were conducted for different setups using our constructed corpus.

The first setup consists of about 260 out of 520 documents which, were tested for most of the methods apart from Sentiment Scoring, where the different amount of documents were used as there was no training phase for Sentiment Scoring. For sentiment analysis, total of 126 documents were treated as Accident, 138 documents as Economics, 94 documents as Entertainment and 102 documents as Politics. Apart from that, rest of the models used 76 documents for Accident, 88 documents for Economics, 44 documents for Entertainment, and 52 documents for Politics. The results are shown in Table 3 and Table 4.

The second setup consists of about 364 out of 520 documents tested for all of the methods. Only 156 documents were used for training purposes.

Sentiment Scoring was not considered this time, and the rest of the models used 106 documents for Accident, 130 documents

Table 3: AVERAGE ROUGE-1 SCORES IN 1ST SETUP

Method Name	Scoring Criteria		
	Precision	***Recall***	***F-Measure***
Keyword Ranking	0.6511	0.6469	0.6402
Sentiment Scoring	0.5706	0.7227	0.6266
Text Ranking	0.5743	0.7045	0.6247
Hybrid 1	0.6501	0.7387	0.6842
Hybrid 2	0.6645	0.7357	0.6907
Hybrid 3	0.6636	0.7156	0.6803

Table 4: AVERAGE ROUGE-2 SCORES IN 1ST SETUP

Method Name	Scoring Criteria		
	Precision	***Recall***	***F-Measure***
Keyword Ranking	0.5565	0.5592	0.5494
Sentiment Scoring	0.4849	0.6183	0.5327
Text Ranking	0.4706	0.6085	0.5230
Hybrid 1	0.5688	0.6529	0.6005
Hybrid 2	0.5861	0.6584	0.6125
Hybrid 3	0.5802	0.6331	0.5974

for Economics, 55 documents for Entertainment, and 73 documents for Politics. The evaluation results of the second setup are shown in Table 5 and Table 6.

Table 5: AVERAGE ROUGE-1 SCORES IN 2ND SETUP

Method Name	Scoring Criteria		
	Precision	***Recall***	***F-Measure***
Keyword Ranking	0.5203	0.5586	0.5253
Text Ranking	0.4783	0.6321	0.5331
Hybrid 1	0.5124	0.6242	0.5508
Hybrid 2	0.5161	0.6124	0.5480
Hybrid 3	0.5197	0.6005	0.5443

Table 6: AVERAGE ROUGE-2 SCORES IN 2ND SETUP

Method Name	Scoring Criteria		
	Precision	***Recall***	***F-Measure***
Keyword Ranking	0.4261	0.4578	0.4308
Text Ranking	0.3808	0.5310	0.4342
Hybrid 1	0.4273	0.5215	0.4600
Hybrid 2	0.4300	0.5120	0.4576
Hybrid 3	0.4320	0.5004	0.4535

In Table 3, Table 4, Table 5, and Table 6, Keyword Ranking refers to the summaries, where only keywords from the documents were considered. In the case of Sentiment Scoring, the summaries were generated, considering only unbiased sentences from the documents. Text Ranking refers to the summaries, where only the most interconnected sentences from the documents were picked. Hybrid 1 refers to the hybrid system where only keyword ranking (40 percent of the total score) and sentiment score (60 percent of the total score) ranking have been used. Hybrid 2 and Hybrid 3 refer to the hybrid systems that combine keyword ranking, sentiment scoring, and text ranking methods. In the Hybrid 2 system, model weights were set to 30, 20, and 50 percent of the total weights for the keyword ranking, sentiment scoring, and text ranking methods, respectively. In the Hybrid 3 system, weights were set to 50, 20, 30 percent, respectively.

From Table 3 and Table 4, we can see that the Hybrid 2 model outperforms all the other models in terms of F-Measure with a ROUGE-1 score of 0.6907 and ROUGE-2 score of 0.6125. Summaries generated by using the Text Ranking method performed the poorest among all of the models.

From Table 5 and Table 6, we can see that the Hybrid 1 model outperforms all the other models in terms of F-Measure with a ROUGE-1 score of 0.5508 and ROUGE-2 score of 0.4600. Summaries generated by using the Keyword Ranking method performed the poorest among all of the models.

From all of the above-mentioned methods Hybrid 2 model from 1st setup achieved the highest score. So, the Hybrid 2 system has been used for conducting further comparison experiments.

Table 7 refers to a detailed breakdown of Table 3 and Table 4. In Table 7, all of the six different models were considered for all of the four classes: Accident, Economics, Entertainment, and Politics. Both ROUGE-1 and ROUGE-2 scores were calculated. Also, the web-based system of Chandro et al. [29] has been considered, where a total of 260 documents of the manual dataset [23], 200 Prothom Alo news documents, and 60 Kaler Kantho news documents were tested and evaluated with the benchmark summaries.

We can see from Table 7, Politics category using the Hybrid 2 model, has the highest ROUGE-1 F-Measure score of 0.7449 and ROUGE-2 F-Measure score of 0.7449. On the other hand, the Entertainment category using the Sentiment Scoring model has the lowest ROUGE-1 F-Measure score of 0.5788 and ROUGE-2 F-Measure score of 0.4870. Due to the nature of Politics documents, the number of documents tested and the summaries generated, it conquered in almost every model. Accident category, despite being more accurately classified, had the best F-Measure score of 0.6950 when used in the Hybrid 2 model. The number of documents used for testing also played a key role here as the Politics category used only 52 documents, whereas the Accident category used 76 news documents. The ROUGE scores for the Entertainment category were very poor in most of the cases. It was expected because most of the news used for the entertainment category was interviews. Generally, interviews do not offer an ideal situation for the summarizer to summarize the document. Even it is hard for humans to summarize interviews. We have analyzed that collecting news documents of other types

Table 7: BREAKDOWN OF ROUGE SCORES IN 1ST SETUP

Method Name	Category	ROUGE-1			ROUGE-2		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
Keyword Ranking	Accident	0.6681	0.6078	0.6293	0.5777	0.5320	0.5470
	Economics	0.6271	0.6650	0.6360	0.5306	0.5696	0.5414
	Entertainment	0.6036	0.5774	0.5804	0.4964	0.4804	0.4783
	Politics	0.7069	0.7322	0.7136	0.6201	0.6477	0.6266
Sentiment Scoring	Accident	0.5652	0.7144	0.6141	0.6240	0.4824	0.5258
	Economics	0.5493	0.7356	0.6199	0.6210	0.4643	0.5231
	Entertainment	0.5291	0.6717	0.5788	0.4456	0.5598	0.4870
	Politics	0.6456	0.7626	0.6946	0.5518	0.6612	0.5964
Text Ranking	Accident	0.6059	0.7059	0.6451	0.5042	0.6261	0.5515
	Economics	0.5491	0.7257	0.6161	0.4476	0.6209	0.5120
	Entertainment	0.5479	0.6444	0.5848	0.4358	0.5320	0.4724
	Politics	0.5930	0.7169	0.6429	0.4896	0.6260	0.5421
Hybrid 1	Accident	0.6740	0.7134	0.6873	0.5977	0.6387	0.6118
	Economics	0.6114	0.7371	0.6589	0.5289	0.6460	0.5731
	Entertainment	0.6346	0.6346	0.6729	0.5492	0.6387	0.5837
	Politics	0.6936	0.7854	0.7318	0.6450	0.6972	0.6450
Hybrid 2	Accident	0.6902	0.7110	0.6950	0.6172	0.6446	0.6251
	Economics	0.6634	0.7309	0.6634	0.5416	0.6466	0.5807
	Entertainment	0.6430	0.7211	0.6733	0.5628	0.6411	0.5919
	Politics	0.7128	0.7922	0.7449	0.6355	0.7130	0.6649
Hybrid 3	Accident	0.6918	0.6821	0.6802	0.6148	0.6131	0.6075
	Economics	0.6219	0.7159	0.6556	0.5358	0.6273	0.5693
	Entertainment	0.6434	0.6990	0.6627	0.5516	0.6058	0.5693
	Politics	0.7098	0.7780	0.7371	0.6288	0.6952	0.6537
Chandro et al. [29]	Prothom Alo	0.6220	0.4019	0.4652	0.5249	0.3253	0.3769
	Kaler Kantho	0.6963	0.4684	0.5405	0.5986	0.4025	0.4614

from this category could improve the score. The overall scores were improved after we have included some short stories, which hint that, the nature of the document plays a vital role in summarizing documents.

We have found that from our experiment, some sentences are more likely to be picked by people if they contain some special keywords. Keywords that appear in the headlines have numerical values, contain names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc., put extra value on a sentence. But in the experiment, we have treated all the keywords equally. Khaleghi et al. [30] have discussed this issue in their paper. They used "Named Entity Recognition" for summarizing Persian text documents. Introducing this for Bangla text in the system may be helpful. Putting extra weight on these keywords should improve the overall ROUGE scores.

4.3. Comparison with existing system models

4.3.1. Comparison with existing web-based summarizer

The web-based system of Chandro et al. [29] has been considered where a total of 260 documents of the manual data set [23] were tested and evaluated with the benchmark summaries.

Following tables compare the ROUGE Scores between both of the models:

Table 8: COMPARISON OF ROUGE-1 SCORES

Method Name	**Scoring Criteria**		
	Precision	***Recall***	***F-Measure***
Proposed Model	0.6645	0.7357	0.6907
Chandro et al. [29]	0.6440	0.4216	0.4661

Table 9: COMPARISON OF ROUGE-2 SCORES

Method Name	**Scoring Criteria**		
	Precision	***Recall***	***F-Measure***
Proposed Model	0.5861	0.6584	0.6125
Chandro et al. [29]	0.5670	0.3481	0.4019

From Table 8 and Table 9, it can be observed that the ROUGE-1 and ROUGE-2 scores of our proposed system model outperformed the ROUGE-1 and ROUGE-2 scores of the web-based system of Chandro et al. [29]. The reason behind the failure of their system can be found by analyzing Table 7. The generated summaries of their system were too short and thus, missing many important sentences to be picked in the summary. Therefore, the Recall score was too low, reducing the overall F-Measure score.

4.3.2. Comparison with other similar systems

In this case, the BNLPC dataset [27] have been considered for the comparison purpose. These dataset have been used previously by Sarkar [7] and also Haque et al. [11] in their summarizers. Machine-generated summaries were compared to 3 different human-generated summary models for each document, and only the average ROUGE scores were considered. The following tables show the average ROUGE scores of our proposed system and their published systems.

Table 10: COMPARISON OF ROUGE-1 SCORES

Method Name	**Scoring Criteria**		
	Precision	***Recall***	***F-Measure***
Proposed Model	0.5658	0.7745	0.6487
Haque et al.[11]	0.5757	0.6819	0.6166
Sarkar [7]	0.5603	0.5515	0.5496

From Table 10, it can be observed that, in the case of ROUGE-1 score, the F-Measure score and Recall score of the proposed system performs better than all of the previously existed models substantially.

Table 11: COMPARISON OF ROUGE-2 SCORES

Method Name	**Scoring Criteria**		
	Precision	***Recall***	***F-Measure***
Proposed Model	0.4958	0.7065	0.5777
Haque et al. [11]	0.5459	0.6433	0.5830
Sarkar [7]	0.5165	0.5075	0.5060

From Table 11, we can see that our ROUGE-2 Recall score is better than all of the existing systems. Our F-Measure score is closer to the one of Haque et al. [11].

To analyze the reason behind low Precision score in both ROUGE-1 and ROUGE-2, it was found that, the benchmark summaries were too short. We considered 40 percent of a document to create an ideal summary, which will give a better perspective of the overall document, but the summaries used in the BNLPC [27] dataset were lower than 40 percent of the actual document. The low precision score also reduced the F-Measure score as it combines both Recall and Precision score.

4.3.3. Comparison with other similar systems for the English language

We undertook an analytical review of our research and contrasted it with 3 extractive summarization methods used for the English language. All of the models used DUC 2002 dataset. We used our manually processed corpus for the Bengali language for the comparison.

The ROUGE Score comparison are shown in the following tables:

Table 12: COMPARISON OF ROUGE-1 SCORES

Method Name	Scoring Criteria		
	Precision	Recall	F-Measure
Uçkan and Karcı [20]	0.4774	0.5195	0.4973
Mutlu et al. [21]	0.6547	0.6629	0.6587
Joshi et al. [22]	-	-	0.5170
Proposed Model	0.6645	0.7357	0.6907

Table 13: COMPARISON OF ROUGE-2 SCORES

Method Name	Scoring Criteria		
	Precision	Recall	F-Measure
Uçkan and Karcı [20]	0.2271	0.2470	0.2365
Mutlu et al.[21]	0.5830	0.5908	0.5869
Joshi et al. [22]	-	-	0.2750
Proposed Model	0.4958	0.7065	0.5777

Since our dataset was formulated for the Bengali language, the contrast with better scores can not specifically determine the process. But we can see that, in terms of the F-Measure score for ROUGE-1, our proposed model performed very well. Our method outperformed every other model in every aspect. In the case of ROUGE-2 F-Measure score, our model is outperformed by the proposed model of Mutlu et al. [21] by a small margin. If we consider the Recall scores, our proposed model outperformed every other model.

4.4. Sample Generated Output

An example of a proposed hybrid model 2 generated summary on a news collected from BNLPC dataset [27].

আজ রোববার দুপুরে গাংনী উপজেলা পরিষদ চত্বরে ভ্রাম্যমাণ আদালত পরিচালনা করে সহকারী কমিশনার (ভূমি) ও নির্বাহী হাকিম রাহাত মান্নান এ দণ্ড দেন।

আজ এ বিয়ের খবর পেয়ে সহকারী কমিশনার (ভূমি) রাহাত মান্নান বর শরীফুল ইসলাম তাঁর শ্বশুর মহিবুল ইসলামকে তাঁর কার্যালয়ে ডেকে পাঠান।

ভ্রাম্যমাণ আদালত সূত্র জানায়, এক মাস আগে গোপনে ব্রজপুর গ্রামের শরিফুলের সঙ্গে মহিবুল ইসলামের দশম শ্রেণি পড়ুয়া মেয়ের বিয়ে হয়।

প্রশাসন গতকাল শনিবার মেহেরপুর জেলা স্টেডিয়ামে গণসমাবেশ করে এই জেলাকে বাল্যবিবাহমুক্ত বলে ঘোষণা দেয়।

সহকারী কমিশনার (ভূমি) রাহাত মান্নান সাংবাদিকদের বলেন, জেলাকে বাল্যবিবাহ মুক্ত রাখার লক্ষ্যে অভিযান অব্যাহত থাকবে।

The subsequent Benchmark Summary collected from BNLPC [27].

মেহেরপুরের গাংনী উপজেলার ব্রজপুর গ্রামে জামাই ও শ্বশুরকে এক মাস করে কারাদণ্ড দিয়েছেন আদালত

ভ্রাম্যমাণ আদালত সূত্র জানায়, এক মাস আগে গোপনে ব্রজপুর গ্রামের

শরিফুলের সঙ্গে মহিবুল ইসলামের দশম শ্রেণি পড়ুয়া মেয়ের বিয়ে হয়

সহকারী কমিশনার (ভূমি) রাহাত মান্নান সাংবাদিকদের বলেন, জেলাকে বাল্যবিবাহ মুক্ত রাখার লক্ষ্যে অভিযান অব্যাহত থাকবে

4.5. Time Analysis for Summary Generation

None of the existing systems performed or presented the time required to generate a summary text. We thought, it will be useful to publish our experimental result to give an idea about time analysis with future directions to improve it.

For our both setups, ten random test documents from each category have been selected for observing the time needed to generate summary for the hybrid summarizer (model 2) and average time per unit time (in seconds) for summary generation has been presented. The configuration used for the testing is as follows: Intel Core i5 Processor, 8 GB Ram and 256 GB SSD.

Table 14: TIME NEEDED TO GENERATE SUMMARY

Setup No	**Category**			
	Acc.	***Eco.***	***Ent.***	***Pol.***
First	0.603	19.362	2.416	17.501
Second	22.562	12.158	5.245	22.068

From Table 14, it can be seen that our hybrid system (Model 2) takes a good amount of time for some classes of documents (based on document types and the length of the actual document). So, naturally, the process was a bit slow in those cases. To improve the model and reduce the time that is taken to generate a summary, a multi-threaded system can be used.

5. CONCLUSION AND FUTURE WORKS

Here in this paper, the detailed design and evaluation steps of the proposed model have been discussed. A hybrid method combining three individual ranking systems has been proposed for generating automatic extractive summaries of Bengali documents. A literature review of several related works on automatic Bengali text summarization techniques has been kept in view. Broad explanations of the workflow along with the evaluation results comparing with benchmark summaries for different modules and also the comparison with other existing systems have been shown. The time needed to generate summaries for different categories has also been analyzed and listed on tables. After evaluating based on ROUGE-1 and ROUGE-2 evaluation measures, the proposed system has shown satisfactory results.

The current work has been presented as an outcome of ongoing research, and in the future, our goal is to train the summarizer for many other different categories like literature, international, editorial, etc. and provide more importance on sentences that are based on headlines and numerical values. Introducing "Named Entity Recognition" for Bangla text in the system will also be helpful in finding important sentences. The Multi-threaded system can be used in order to reduce the summary generation time.

References

[1] K. Sarkar, Bengali text summarization by sentence extraction, in: Proceedings of International Conference on Business and Information Management(ICBIM-2012), 2012, pp. 233--245.

[2] M. N. Uddin, S. A. Khan, A study on text summarization techniques and implement few of them for bangla language, in: 2007 10th international conference on computer and information technology, IEEE, 2007, pp. 1--4.

[3] A. Das, S. Bandyopadhyay, Topic-based bengali opinion summarization, in: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Association for Computational Linguistics, 2010, pp. 232--240.

[4] R. Mihalcea, P. Tarau, Textrank: Bringing order into text, in: Proceedings of the 2004 conference on empirical methods in natural language processing, 2004, pp. 404--411.

[5] R. Mihalcea, Graph-based ranking algorithms for sentence extraction, applied to text summarization, in: Proceedings of the ACL Interactive Poster and Demonstration Sessions, 2004, pp. 170--173.

[6] K. Sarkar, An approach to summarizing bengali news documents, in: proceedings of the International Conference on Advances in Computing, Communications and Informatics, 2012, pp. 857--862.

[7] K. Sarkar, A keyphrase-based approach to text summarization for english and bengali documents, International Journal of Technology Diffusion (IJTD) 5 (2) (2014) 28--38.

[8] N. Srivastava, B. K. Gupta, An algorithm for summarization of paragraph up to one third with the help of cue words comparison, International Journal of Advanced Computer Science and Application (IJACSA), O ISSN (2014) 2156--5570.

[9] P. Chandro, M. F. H. Arif, M. M. Rahman, M. S. Siddik, M. S. Rahman, M. A. Rahman, Automated bengali document summarization by collaborating individual word & sentence scoring, in: 2018 21st International Conference of Computer and Information Technology (ICCIT), IEEE, 2018, pp. 1--6.

[10] M. I. A. Efat, M. Ibrahim, H. Kayesh, Automated bangla text summarization by sentence scoring and ranking, in: 2013 International Conference on Informatics, Electronics and Vision (ICIEV), IEEE, 2013, pp. 1--5.

[11] M. M. Haque, S. Pervin, Z. Begum, Enhancement of keyphrase-based approach of automatic bangla text summarization, in: 2016 IEEE Region 10 Conference (TENCON), IEEE, 2016, pp. 42--46.

[12] S. Mandal, G. K. Singh, A. Pal, Pso-based text summarization approach using sentiment analysis, in: Computing, Communication and Signal Processing, Springer, 2019, pp. 845--854.

[13] R. K. Roul, J. K. Sahoo, Sentiment analysis and extractive summarization based recommendation system, in: Computational Intelligence in Data Mining, Springer, 2020, pp. 473--487.

[14] N. Yadav, N. Chatterjee, Text summarization using sentiment analysis for duc data, in: 2016 International Conference on Information Technology (ICIT), IEEE, 2016, pp. 229--234.

[15] M. M. Haque, S. Pervin, Z. Begum, Automatic bengali news documents summarization by introducing sentence frequency and clustering, in: 2015 18th International Conference on Computer and Information Technology (ICCIT), IEEE, 2015, pp. 156--160.

[16] J. Li, G. Huang, C. Fan, Z. Sun, H. Zhu, Key word extraction for short text via word2vec, doc2vec, and textrank, Turkish Journal of Electrical Engineering & Computer Sciences 27 (3) (2019) 1794--1805.

[17] Y.-H. Hu, Y.-L. Chen, H.-L. Chou, Opinion mining from online hotel reviews--a text summarization approach, Information Processing & Management 53 (2) (2017) 436--449.

[18] S. Basheer, M. Anbarasi, D. G. Sakshi, V. V. Kumar, Efficient text summarization method for blind people using text mining techniques, International Journal of Speech Technology (2020) 1--13.

[19] C. Xiong, X. Li, Y. Li, G. Liu, Multi-documents summarization based on textrank and its application in online argumentation platform, International Journal of Data Warehousing and Mining (IJDWM) 14 (3) (2018) 69--89.

[20] T. Uçkan, A. Karcı, Extractive multi-document text summarization based on graph independent sets, Egyptian Informatics Journal (2020) 1--13.

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

[22] A. Joshi, E. Fidalgo, E. Alegre, L. Fernández-Robles, Summcoder: An

unsupervised framework for extractive text summarization based on deep auto-encoders, Expert Systems with Applications 129 (2019) 200--215.

[23] Mahimul, Hybrid-Text-Summarizer-For-Bangla-Document, https://github.com/mahimulislam/ResourcesHybridTextSummarization (2020 (accessed May 30, 2020)).

[24] Imran, Bengali-Sentiment-Analysis, https://github.com/Imran-cse/Bengali-Sentiment-Analysis (2018 (accessed August 5, 2019)).

[25] Y. Chen, S. Skiena, Building sentiment lexicons for all major languages, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), 2014, pp. 383--389.

[26] A. R. Deshpande, L. Lobo, Text summarization using clustering technique, International Journal of Engineering Trends and Technology 4 (8) (2013) 3348--3351.

[27] B. N. L. P. Community, Dataset for Evaluating Bangla Text Summarization System, https://www.bnlpc.org/research.php (2016 (accessed August 5, 2019)).

[28] C.-Y. Lin, E. Hovy, 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, 2003, pp. 150--157.

[29] Porimol, Web Based Bengali Document Summarizer, https://bengali-document-summarizer.herokuapp.com (2018 (accessed May 30, 2020)).

[30] Z. Khaleghi, M. Fakhredanesh, M. Hourali, Mscso: Extractive multi-document summarization based on a new criterion of sentences overlapping, Iranian Journal of Science and Technology, Transactions of Electrical Engineering (2020) 1--11.