Unsupervised Bengali Text Summarization Using Sentence Embedding and Spectral Clustering

Sohini Roy Chowdhury and **Kamal Sarkar** and **Arka Maji**
jukamal2001@yahoo.com
Department of Computer Science and Engineering
Jadavpur University
188, Raja S.C. Mallick Rd
Kolkata-700032,West Bengal
India

Abstract

Single document extractive text summarization produces a condensed version of a document by extracting salient sentences from the document. Most significant and diverse information can be obtained from a document by breaking it into topical clusters of sentences. The spectral clustering method is useful in text summarization because it does not assume any fixed shape of the clusters, and the number of clusters can automatically be inferred using the Eigen gap method. In our approach, we have used word embedding-based sentence representation and a spectral clustering algorithm to identify various topics covered in a Bengali document and generate an extractive summary by selecting salient sentences from the identified topics. We have compared our developed Bengali summarization system with several baseline extractive summarization systems. The experimental results show that the proposed approach performs better than some baseline Bengali summarization systems it is compared to.

1 Introduction

With the advancement of search engines, we are flooded with information. This information overload problem affects the proficiency of decision-making of humans. Instead of time waste, it also affects the capacity of humans. In today's world where each day technology is changing our daily life, the human brain plays an important role there. So it is unworthy to waste the human brain and time in a negative way. Having a crux with relevant information from a long document manually is a very tedious task. Text summarization is a very useful solution to this information overload problem. Text summarization helps to create a condensed version of a document by selecting sentences with pertinent information from the document. Text summarization demands well understanding of the document to create the gist. Text summarization can be categorized into two types: extractive and abstractive. Extractive summarization aims to generate a summary by selecting textual segments or sentences from the document whereas abstractive summaries are generated from the document by introducing new words or phrase which may not be present in the original document. Although abstractive summaries are more human-like than extractive summaries, the state-of-the art abstractive summarization approaches are not good enough in producing an abstract from a longer document. Many existing abstractive summarization approaches use two step process. in the first step, an extract is generated. and in the next step, an abstract is generated by reformulating the sentences in the extract(Sarkar, 2010). Thus the extractive summarization is useful. Moreover, Bengali is a resource-scarce language and abstractive summarization requires a large amount of language resources which are not available for Bengali language. This motivates us to work on Bengali extractive summarization.

Capturing connectivity among sentences of a document is helpful to group similar sentences and create a condensed extract. Sentence clustering is an unsupervised method that groups similar sentences and produces clusters. Traditional clustering algorithms though widely used earlier have some pitfalls which are overcome using spectral clustering. Spectral clustering emphasizes creating more accurate clusters than traditional clustering algorithms as it does not make assumptions about the shape of the cluster. Spectral clustering utilizes the connectivity of data points. If two data points appear side by side but are not connected, spectral clustering will not group them together. The main benefits of using spectral clustering in document segmentation are that the clusters produced by this method do not follow any fixed shape. We assume that the clusters representing topics are non-Gaussian. We consider that the spectral clustering algorithm is suitable for segmenting a

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 337 - 346

document into multiple topical clusters where a cluster represents a topical segment that consists of semantically similar sentences appearing in close positional proximity.

In our approach, each sentence vector is computed by averaging the word embedding vectors obtained using fasttext [1] open source. After obtaining the sentence vector by averaging the word vectors, the position of the sentence is included as an additional dimension of the sentence vector. Position information is considered to encourage locally coherent sentences to fall in the same cluster. Clusters are ranked according to the average position number of the sentences in the cluster and a summary is created by choosing the most relevant sentences from the ranked clusters. Sentence selection from the cluster is also done in an effective way. The efficacy of our approach lies in the effectiveness of spectral clustering in segmenting the document into multiple topical clusters.

The approach proposed in this paper differs from the existing approaches (Günes and Dragomir R., 2004)(Sarkar, 2009a)(Sarkar, 2009b)(Sarkar, 2008)(Sarkar, 2012a)(Sarkar, 2012b)(Sarkar and Bandyopadhyay, 2005). We use the spectral clustering algorithm to segment a document into multiple topical clusters and create a summary by choosing topic-wise most relevant sentences. On the other hand, the existing approaches (Günes and Dragomir R., 2004)(Sarkar, 2009a)(Sarkar, 2009b)(Sarkar, 2008)(Sarkar, 2012a)(Sarkar, 2012b)(Sarkar and Bandyopadhyay, 2005) decompose the entire document into a collection of sentences and rank the sentences based on some features to create a summary. So, our proposed approach uses an effective clustering-based method that produces a summary covering all important topics in a document.

Our paper is set up in the following manner. Related work is discussed in section 2. Our proposed methodology is explained in section 3. Section 4 highlights the dataset used in the approach results and comparison among existing models. Section 5 concludes the paper.

2 Related Work

In the area of extractive text summarization, the early approaches used various heuristic algorithms to identify important segments from a document. The methods that include features like sentence position, word frequency, and key phrases to extract salient sentences from the document have been presented in (Baxendale, 1958)(Edmundson, 1969)(Luhn, 1958). Most early text summarization algorithms faced the redundancy problem or the diversity problem. So, to deal with these problems and assuring good coverage, clustering of sentences is used (Jain and Dubes, 1988). The idea of employing a clustering algorithm for text summarization was well described in (Sarkar, 2009a). This approach used three steps for text summarization: histogram-based clustering algorithm for sentence clustering, ordering of clusters, and extraction of summary-worthy sentences from the clusters to create the summary.

In (Jing and McKeown, 2000), a hierarchical agglomerative clustering algorithm was used to create clusters of sentences. To create a summary, sentences were chosen in order from largest to smallest cluster. Another clustering-based approach presented in (Wan and Yang, 2008) incorporates cluster-level information in a graph model for ranking sentences.

However, the early works (Sarkar, 2009a) suggest that the performance of clustering-based text summarization heavily depends on the quality of clusters produced. Clustering algorithms perform well when we have a clear idea regarding attributes of data points (Jin, 2006). Clustering based on compactness highlights spatial proximity among data points. For example, agglomerative average link clustering (Jain and Dubes, 1988), k-means (Hartigan and Wong, 1979), highlights compactness. The resultant clusters using this algorithm is spherical clusters. Modification on k-means was discussed in (Arthur and Vassilvitskii, 2007) which is defined as k-means++. Though it uses a better centroid initialization technique for improvements over k-means, still it suffers from some drawbacks because we need to specify the number of clusters to be formed in advance and it assumes a fixed shape of clusters. After investigating different existing clustering algorithms, we can find that spectral clustering is more suitable for our task. It embeds sentences on a low-dimensional eigen space and performs clustering on the data points mapped to the low-dimensional embedding space. It does not assume any fixed shape of the cluster but rather emphasizes graph partitioning (Hamad and Biela, 2008) based

[1] https://fasttext.cc/docs/en/crawl-vectors.html

on connectedness among the vertices representing the data points. So, it is very useful when the shape of cluster is non-convex. Nowadays, the spectral clustering algorithm has been used in a wide range of application areas like image clustering (Tilton, 1998), shape clustering (Sidi et al., 2011), motion clustering (Lauer and Schnörr, 2009) and many more. Gupta et. al. (Gupta et al., 2019) presented a spectral clustering-based text summarization approach, which uses Textual Entailment(TE) and Spectral Clustering (ATESC) to calculate sentence connectedness scores. It is used to measure the saliency of a sentence in the input.
However, to the best of our knowledge, it is our new attempt to use a spectral clustering algorithm in the Bengali text summarization domain. For sentence representation, we have also considered sentence position as a new feature and combined it with the semantic content-based sentence features. The spectral clustering is applied to the sentence vectors to produce multiple clusters where each cluster represents a topical segment of the input document. The final summary is generated by choosing sentences from the ordered clusters using a centrality-based saliency measure (Günes and Dragomir R., 2004).

3 Our Proposed Methodology

Steps of our proposed system is illustrated in Figure 1. Each step of the proposed system is discussed in this section.

3.1 Preprocessing

Sentences are identified using a sentence tokenizer available with the NLTK toolkit. A sentence is split up into words. Stop words are discarded from the sentences. Stop words denotes unimportant frequent words in the dataset. A predefined, human-made list of Bengali words [2] was considered for stop word removal. 363 stop words were considered in that stop word list. A sample sentence after discarding stop words from it is represented in Figure 2.

3.2 Sentence Vectorization

After pre-processing, sentences are passed to the vectorization step. The vector for a sentence is obtained by taking an average of the vectors corresponding to the words that appeared in the sentence.

[2]http://fire.irsi.res.in/fire/static/resources

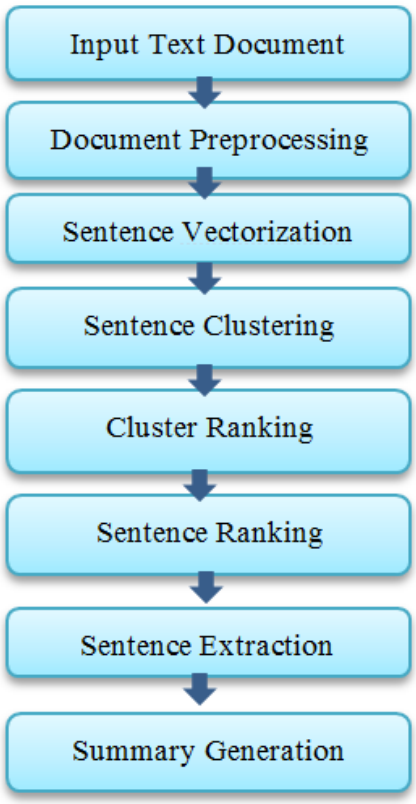

Figure 1: Steps of the proposed summarization system

Sample sentence:
ইডেনে গমগম করে উঠল ঢাকের আওয়াজ
After removing stop word from sentence:
ইডেনে গমগম উঠল ঢাকের আওয়াজ

Figure 2: Removal of stop word for Bengali sentence

fastText word embeddings [3] were used to get word vectors. Since the size of a word vector is 300, the dimension of the sentence vector obtained using the average rule is 300. The value for the feature "sentence position" is appended at the end of the sentence vector, which increases its dimension to 301. The value for the sentence position feature is calculated as the division of the position of the sentence in the document by the total sentences in the document. Hence our final sentence vector is of dimension 301. The rationale behind including sentence position in the sentence vector is to encourage locally coherent sentences to fall in the same cluster. This helps to segment a document in a better way.

[3]https://fasttext.cc/docs/en/crawl-vectors.html

3.3 Sentence Clustering

In this step, sentence vectors are clustered into clusters of different sizes. The idea is to group similar and closer sentences into the same cluster. To cluster the sentences, we have used the spectral clustering algorithm. To implement spectral clustering, we first calculate the affinity matrix from a document graph in which a node corresponds to a sentence vector, and the edge between two nodes is weighted by the similarity between the corresponding two vectors. Affinity matrix is created using the similarity function given in Equation 1, which is basically a Gaussian similarity function.

$$A_{ij} = exp(\frac{-d^2(s_i, s_j)}{\sigma^2}) \quad (1)$$

Where σ is a control parameter that controls the context window in our case. In equation 1, $d(s_i, s_j)$ denotes distance between two sentence vectors s_i and s_j. Distance between two points (x_1, x_2) and (y_1, y_2) is calculated using the formula of Euclidean Distance defined in equation 2. In our approach, we varied $sigma$ and got the best result when it is set to 10.

$$dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

From the affinity matrix, the graph Laplacian matrix is obtained using equation 3.

$$L = D - A \quad (3)$$

where A is the affinity matrix and D is the degree matrix such that

$$d_i = \sum_{j|(i,j)\in E} W_{ij} \quad (4)$$

where E is the set of edges in the graph and W_{ij} refers to the similarity between two points x_i and x_j corresponding to two different sentences in a document.

After normalizing the graph Laplacian matrix, the Eigen values and Eigen vectors of the normalized graph Laplacian matrix are used to embed the sentences into a low dimension Eigen space (Luxburg, 2007). Finally, a simple k-means clustering algorithm is applied for clustering the low dimensional dense vectors to obtain hard clusters. The main problem in the K-means cluster algorithm is that it needs to specify the value of K in advance. In our case, we have used an Eigen map heuristic method to determine the value of K. The main idea is to choose the value K such that all eigenvalues λ_1,.....λ_k are very small, but λ_{k+1} is relatively large. The details of this method can be found in (Luxburg, 2007). We have used this method to determine the number of clusters. Thus the number of topical segments is automatically inferred in an unsupervised way.

3.4 Cluster Ranking

The clusters are ranked in ascending order on the basis of the average of the position values of the sentences, present in that cluster. The cluster ranking enables us to identify the more significant clusters from which sentence extraction will occur first. The rationale behind using the position-based cluster ranking method is to ensure the selection of sentences in the summary from the topics in order as they appear in the text (position-based topical order). This is useful in creating an informative extract of sentences covering various topics in a document.

3.5 Within-cluster Sentence Ranking

A particular cluster may have multiple sentences present in it. To identify the most salient sentence from each cluster, the sentences within a particular cluster are ranked using the graph-based lexical centrality method published in (Günes and Dragomir R., 2004). In this method, a weighted adjacency matrix is constructed for the graph representing each cluster where the sentences in the cluster are considered as the vertices and the cosine similarity is considered as the edge weights between two sentence vectors. Cosine similarity is one of the popular similarity measures between two vectors. It is the cosine of the angle between two vectors, which means the dot product of two vectors divided by the product of their lengths. Cosine similarity is calculated using equation 5, where A and B are two sentence vectors belonging to a cluster. The rank of the sentence in a particular cluster is the sum of all the cosine edge weights to all other vertices in the cluster graph. The higher the sum of edge weights the higher the rank of the sentence is. This score is used to identify the sentence which is the most central to the cluster.

$$cosine - similarity = \frac{A.B}{||A||||B||} \quad (5)$$

For example, let us assume that six sentences are present in a cluster. Now adjacency matrix is created for that cluster using the cosine similarity

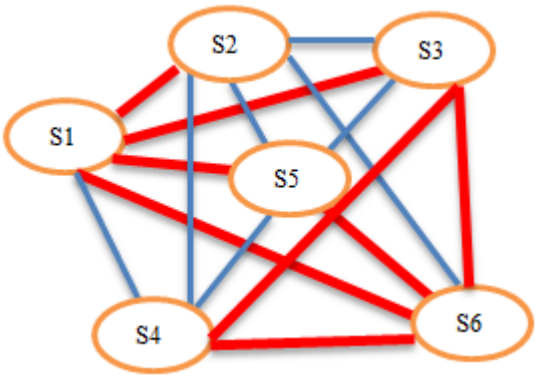

Figure 3: A sample similarity graph for the sentences in a cluster

value. A sample weighted matrix for the graph representing a cluster is as follows.

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \end{array} \begin{array}{c} \begin{array}{cccccc} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \end{array} \\ \begin{pmatrix} 0 & 0.8 & 0.7 & 0.3 & 0.8 & 0.75 \\ 0.8 & 0 & 0.3 & 0.2 & 0.4 & 0.3 \\ 0.7 & 0.3 & 0 & 0.6 & 0.3 & 0.7 \\ 0.3 & 0.2 & 0.6 & 0 & 0.2 & 0.8 \\ 0.8 & 0.4 & 0.3 & 0.2 & 0 & 0.9 \\ 0.75 & 0.3 & 0.7 & 0.8 & 0.9 & 0 \end{pmatrix} \end{array}$$

In the above sentence similarity matrix, the row sum gives the sum of the similarities of a given sentence to the other sentences in the cluster they belong to. From the above sentence similarity matrix, we can observe that s_6 has the highest saliency score and it should be selected first as a summary worthy sentence from the cluster.

3.6 Summary Generation

After cluster ordering, sentences within each cluster are ranked. Then the sentence extraction process begins to create a summary for each input document. Here we select the first-ranked sentence in the first cluster followed by the first-ranked sentence in the second cluster and so on. If the number of clusters produced by the clustering algorithm is less than the required number of sentences, the process is repeated in a round-robin fashion until we obtain the required number of sentences. Once the required number of sentences is extracted to attain the desired summary length, the process is stopped. In our approach, a summary of 100 words is taken for evaluation. The algorithm for the overall summary generation process is shown in Algorithm 1.

Algorithm 1 Summary generation using spectral clustering based document segmentation

Input: A text document.

Output: Summary of the document.

1: Breaking the input document into sentences.
2: Removal of stop words from the sentences.
3: Calculation of the 300-dimensional sentence vector by taking an average of word vectors obtained using the fastText open source.
4: Calculate the positional feature value for each sentence and append it with the 300-dimensional sentence vector obtained in the previous step. The positional feature value is calculated as the position of the sentence in the document divided by the total number of sentences in the document.
5: Sentence affinity matrix is created using the Gaussian similarity function.
6: Compute the normalized graph Laplacian for the sentence affinity matrix.
7: Eigenvalue decomposition is performed on the normalized graph Laplacian to get eigenvalues and eigenvectors. The eigenvectors are arranged in the ascending order of eigenvalues.
8: Take the first d eigenvectors to form an N × d matrix U. A matrix T is obtained from U by normalizing the rows of U to norm 1.
9: Eigen gap heuristic is applied to identify the gap which gives the optimal number of clusters, K.
10: Treating each row of T as a spectral embedding of a sentence, a simple K-means algorithm is applied on T to obtain K clusters where K is computed using the Eigen gap heuristic.
11: Clusters are ranked on the basis of the average position values of the sentences belonging to a cluster.
12: The sentences within each cluster are assigned scores based on the graph centrality-based saliency measure (Günes and Dragomir R., 2004).
13: For creating a summary, the top-ranked cluster contributes first its best sentence to the summary and then the second-ranked cluster contributes, and so on. If the number of clusters produced by the clustering algorithm is less than the required number of sentences, the process is repeated in a round-robin fashion until we obtain the required number of sentences.

4 Datasets and Experimental Results

Since no publicly available dataset is available for Bengali text summarization, we have tested our proposed approach on our own dataset consisting of 102 Bengali document-summary pairs. The average number of sentences in each document is 40. For evaluation, we have taken 100 words from each summary generated by the system using the proposed approach.

We have conducted six experiments to prove the effectiveness of our approach. These experiments include implementations of five state-of-the-art unsupervised methods with which our proposed approach is compared. A brief description of the models implemented by us is given below in this section.

Model 1: This is our proposed model that uses clustering-based document segmentation for text summarization.

Model 2: This approach was developed by Luhn (Luhn, 1958). It generates a summary from a document by considering that the sentence containing more frequent words is more important than the sentence containing less frequent words. In this method, stop words are removed before sentence weight calculation.

Model 3: This approach was developed by (Günes and Dragomir R., 2004). It uses Lexrank, which is a graph-based approach that represents sentences as vertices of a graph and considers the cosine similarity between any two sentences as the weight of the edge between the corresponding vertices. Finally, Google's page rank algorithm is applied to the graph to rank sentences.

Model 4: This approach was developed by (Ani and Lucy, 2005). The approach is named "Sumbasic" which considers term frequency as the saliency of a term. Here probability of a word is calculated based on its frequency and each sentence is assigned a score equal to the average probability of the words contained in the sentence. The sentence with the highest score is selected first in the summary. Before selecting the next sentence, the words present in the already selected sentence are penalized by multiplying their probability values by themselves, and the sentences are re-ranked using the newly calculated probability values. After re-ranking the sentences, the sentence with the highest score is selected as the second sentence of the summary. This process is continued until the summary of the desired length is obtained.

Model 5: This model was developed by (Rada and Paul, 2004). It is called as Textrank. It is also a graph-based approach similar to that used in LexRank(Günes and Dragomir R., 2004). LexRank used TF*IDF-based term weight and cosine similarity value as the edge weight whereas TextRank used word overlap-based similarity value as the edge weight.

Model 6: This is the lead baseline model, where a summary is generated by considering the first 100 words of the input document. This is the baseline defined in DUC 2001 and DUC 2002 shared tasks on single document summarization.

4.1 Summary Evaluation Metric

To calculate the performance score of the proposed model, we have used the popular summary evaluation package called ROUGE (Lin, 2004) which measures n-gram overlap between a system-generated summary and the reference summaries (Wan and Yang, 2006). In our case, we have used one reference summary for each system-generated summary. ROUGE counts various kinds of overlapping units between the system summary and the reference summaries. We have used the latest version of the ROUGE package - ROUGE 1.5.5 for evaluating the system summaries. The ROUGE toolkit reports various ROUGE–N scores, for example, ROUGE-1, ROUGE-2, etc. Along with ROUGE-1 scores, many state-of-the-art summarization systems have been evaluated using ROUGE-2 (bigram-based), and ROUGE-SU4 (skip bigrams with skip distance up to 4 words (Lin, 2004)). So, we consider ROUGE-1, ROUGE-2, and ROUGE-SU4 scores for evaluating our proposed summarization models. We set the summary length to 100 words by using the -l 100 option in the ROUGE toolkit, which takes the first 100 words from each system summary for evaluation. We use ROUGE-F score scores to evaluate and compare our proposed neural summarization method with other existing summarization methods.

4.2 Results and Comparisons

We have implemented five existing summarization systems for comparing them with the system proposed by us. The comparison results are shown in Table 1.

As we can see from the table 1, our proposed model (Model 1) performs significantly better than

MODEL	Rouge 1	Rouge 2	Rouge SU4
Model 1(Proposed Model)	**0.4481**	**0.2844**	**0.2848**
Model 2 (Luhn, 1958)	0.3929	0.2324	0.2293
Model 3 (Günes and Dragomir R., 2004)	0.3693	0.1995	0.1980
Model 4 (Ani and Lucy, 2005)	0.3602	0.1831	0.1836
Model 5 (Rada and Paul, 2004)	0.3499	0.1846	0.1835
Model 6(Lead Baseline)	0.2733	0.1501	0.1487

Table 1: Performance of our proposed summarization model and its comparison with some existing summarization methods

other baseline models to which it is compared. The proposed model also performs significantly better than Model 3 (Günes and Dragomir R., 2004) which uses the graph-based ranking of all sentences of the input document considering all sentences in the document as a single cluster. Compared to the system "LexRank " (Günes and Dragomir R., 2004), we use spectral clustering-based document segmentation and within-cluster sentence ranking. It is evident from the results that, instead of taking the input document as a single cluster of sentences, if the document is segmented into multiple topical clusters and the summary is generated by choosing sentences from the clusters one by one, this produces a summary which is better in quality than that produced by the system called "LexRank".

We have computed performance improvement using equation 6 which computes the difference between ROUGE scores obtained by the proposed model and the model to which it is compared.

$$PI = M - N \tag{6}$$

where PI denotes performance improvement, M is the score for the proposed approach, and N is the score for the approach to which the proposed approach is compared. Performance Improvement of our proposed approach over other approaches is shown in Table 2.

MODEL	Rouge 1	Rouge 2
Model 2	0.0552	0.052
Model 3	0.0788	0.0849
Model 4	0.0879	0.1013
Model 5	0.0982	0.0998
Model 6	0.1748	0.1343

Table 2: Performance improvement achieved by our proposed model in comparison with some state-of-the art summarization methods

As we can see from Table 2, the proposed approach shows improvement over the lead baseline and the LexRank(Günes and Dragomir R., 2004) by 0.1748 and 0.0788 ROUGE-1 points respectively.

4.2.1 Comparison of the spectral clustering algorithm with another conventional clustering algorithm

To prove the effectiveness of the spectral clustering algorithm in producing topical segments, we have implemented a variant of the proposed by replacing the spectral clustering algorithm with another popular clustering algorithm called DBSCAN. which is a density-based clustering algorithm(DBSCAN). DBSCAN algorithm is known to be robust to outliers. Minpts(minimum number of points for a cluster) and epsilion are two parameters that are tuned to achieve better performance. The best results are achieved by setting minpts=5 and epsilion=0.4. The summarization evaluation scores are shown in Table 3. It is evident from these results that spectral clustering is more effective for segmenting a document into multiple topics.

MODEL	Rouge 1	Rouge 2
Proposed Model	**0.4481**	**0.2844**
DBSCAN	0.3765	0.2253

Table 3: Comparison of the proposed model with spectral clustering with its variant that uses the DBSCAN clustering algorithm

4.3 An Example

In this subsection, We have shown the clustering and the summarization results for an example input Bengali input document. The clusters produced by the spectral clustering algorithm, a reference summary and the system generated summary are shown in Figure 4 and 5 respectively.

The bold sentences in Figure 4 are the sentences selected by the summarization model proposed by us. Though we have shown in Figure 5 a system

Cluster 1	নিম্নচাপের পরে বর্ষার সঙ্গে জোট ঘূর্ণাবর্তের। **ভিন্ রাজ্যে সরে গিয়েও নিজের ক্ষমতা অটুট রেখে বেশ কয়েক দিন ভুগিয়েছে নিম্নচাপ।** তার প্রভাবে প্রবল বর্ষণের ফলে তৈরি হওয়া বন্যা পরিস্থিতি থেকে পুরোপুরি রেহাই মেলেনি এখনও।
Cluster 2	**এর মধ্যেই হাজির হয়েছে ঘূর্ণাবর্ত।**
Cluster 3	**আর তার জেরে দক্ষিণবঙ্গে মৌসুমি অক্ষরেখা ফের সক্রিয় হয়ে পড়ল।**
Cluster 4	**এর আগে নিম্নচাপ ঠিক এই ভাবেই অতি গভীর নিম্নচাপে রুপান্তরিত হয়ে মৌসুমি অক্ষরেখাকে অতি সক্রিয় করে তুলেছিল।**
Cluster 5	আলিপুর আবহাওয়া দপ্তরের অধিকর্তা গোকুলচন্দ্র দেবনাথ রবিবার জানান , দক্ষিণবঙ্গে কলকাতা এবং সংলগ্ন জেলাগুলির উপরে একটি ঘূর্ণাবর্ত তৈরি হয়েছে। **আজ,সোমবারেও কলকাতা ও পার্শ্ববর্তী এলাকায় বিক্ষিপ্ত বৃষ্টির পূর্বাভাস দিয়েছে আলিপুর আবহাওয়া দফতর।** হাওয়া অফিস জানাচ্ছে, আজ না-হোক, কাল, মঙ্গলবার থেকে উত্তরবঙ্গ ঘেঁষা জেলাগুলিতে বৃষ্টি বাড়বে। ওই ঘূর্ণাবর্ত বা তার সংস্পর্শে ফের সক্রিয় হয়ে ওঠা মৌসুমি অক্ষরেখা ঠিক কী ঘটাতে পারে? সক্রিয় মৌসুমি অক্ষরেখা শনিবার বৃষ্টি নামিয়েছে কলকাতা এবং সংলগ্ন বিভিন্ন জেলায়।
Cluster 6	**রবিবারেও তা মহানগরীর পিছু ছাড়েনি।**
Cluster 7	বাদ যাবে না উত্তরবঙ্গও। **বুধবার থেকে ভারী বৃষ্টি মালদহ দুই দিনাজপুরে।** দক্ষিণবঙ্গের বিভিন্ন জেলায় ভারী বৃষ্টিও হতে পারে বলে জানিয়ে দিয়েছেন আহববিদেরা ।
Cluster 8	**তার পরে উত্তরবঙ্গের জেলাগুলিতে বৃষ্টির দাপট বেড়ে যেতে পারে।**
Cluster 9	**তার প্রভাবেই বর্ষা ফের কিছুটা শক্তিশালী হয়ে উঠেছে।**
Cluster 10	**তাতেই বৃষ্টি হচ্ছে।**
Cluster 11	**তিনি বলেন , "ওই ঘূর্ণাবর্তটি আরও উপরের দিকে উঠে যাবে এবং উত্তরবঙ্গের দিকে সরে যাবে।"** তার পরে সেটি মুর্শিদাবাদ এবং মালদহ হয়ে ঢুকে পড়বে উত্তরবঙ্গে।
Cluster 12	হাওয়া অফিসের পূর্বাভাস অনুযায়ী কলকাতা ও সংলগ্ন এলাকা থেকে ঘূর্ণাবর্তটি আজ , সোমবারেই বর্ধমান হয়ে চলে যাবে বীরভূমের দিকে। তাই ঘূর্ণাবর্তটি যখন যেখানে থাকবে , সেখানে ভাল বৃষ্টি দিয়ে যাবে। এখন গোটা রাজ্যেই মৌসুমি অক্ষরেখা আছে।

Figure 4: Clusters produced using spectral clustering algorithm

Machine Generated Summary

ভিন রাজ্যে সরে গিয়েও নিজের ক্ষমতা অটুট রেখে বেশ কয়েক দিন ভুগিয়েছে নিম্নচাপ ।
এর মধ্যেই হাজির হয়েছে ঘূর্ণাবর্ত ।
আর তার জেরে দক্ষিণবঙ্গে মৌসুমি অক্ষরেখা ফের সক্রিয় হয়ে পড়ল ।
এর আগে নিম্নচাপ ঠিক এই ভাবেই অতি গভীর নিম্নচাপে রুপান্তরিত হয়ে মৌসুমি অক্ষরেখাকে অতি সক্রিয় করে তুলেছিল ।
আজ , সোমবারেও কলকাতা ও পার্শ্ববর্ত এলাকায় বিক্ষিপ্ত বৃষ্টির পূর্বাভাস দিয়েছে আলিপুর আবহাওয়া দফতর ।
রবিবারেও তা মহানগরীর পিছু ছাড়েনি ।
বুধবার থেকে ভারী বৃষ্টি হবে মালদহ এবং দুই দিনাজপুরে ।
তার পরে উত্তরবঙ্গের জেলাগুলিতে বৃষ্টির দাপট বেড়ে যেতে পারে ।
তার প্রভাবেই বর্ষা ফের কিছুটা শক্তিশালী হয়ে উঠেছে ।
তাতেই বৃষ্টি হচ্ছে ।
তিনি বলেন , "ওই ঘূর্ণাবর্তটি আরও উপরের দিকে উঠে যাবে এবং উত্তরবঙ্গের দিকে সরে যাবে ।"

Human Generated Summary

ঘূর্ণাবর্তের বৃষ্টি আজ।
নিম্নচাপ বিদায় নিতে না নিতে হাজির ঘূর্ণাবর্ত।
তার প্রভাবে দক্ষিণবঙ্গে ফের সক্রিয় হয়ে উঠেছে মৌসুমি অক্ষরেখা।
শনিবার থেকে তারই বৃষ্টি চলছে কলকাতা এবং সংলগ বিভিন্ন জেলায়।
আজ সোমবার মহানগরী এবং পার্শ্ববর্ত এলাকায় বিক্ষিপ্ত বৃষ্টির পূর্বাভাস দিয়েছে হাওয়া অফিস।
ভারী বৃষ্টি হতে পারে দক্ষিণবঙ্গের বিভিন জেলায়।
বুধবার থেকে ভারী বৃষ্টি হবে মালদহ ও দুই দিনাজপুরে।
তার পরে উত্তরবঙ্গের জেলাগুলিতে বৃষ্টির দাপট বেড়ে যেতে পারে ।

Figure 5: System generated summary and reference summary of the document

generated summary consisting of 11 sentences, the first 100 words of it is taken during evaluation using the ROUGE package.

5 Conclusion

This paper describes a spectral clustering-based method for segmenting a document into multiple topical segments and a summarization method that generates an extractive summary by choosing sentences from the clusters. Our proposed summarization approach outperforms several existing baseline summarization approaches.

The higher ROUGE score obtained by the proposed approach proves that the spectral clustering algorithm provides more accurate topical segments of a document if the sentence position is added as an additional dimension to the sentence vector obtained by averaging the word vectors.

We have a future plan to use more deep semantic methods for document segmentation and incorporate them into the text summarization process.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Nenkova Ani and Vanderwende Lucy. 2005. The impact of frequency on summarization. *Computer Science*.

David Arthur and Sergei Vassilvitskii. 2007. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA. Society for Industrial and Applied Mathematics.

Phyllis B. Baxendale. 1958. Machine-made index for technical literature - an experiment. *IBM J. Res. Dev.*, 2:354–361.

H. P. Edmundson. 1969. New methods in automatic extracting. *J. ACM*, 16(2):264–285.

Erkan Günes and Radev Dragomir R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

Anand Gupta, Manpreet Kaur, Ahsaas Bajaj, and Ansh Khanna. 2019. Entailment and spectral clustering based single and multiple document summarization. *International Journal of Intelligent Systems and Applications*, 11:39–51.

Denis Hamad and Philippe Biela. 2008. Introduction to spectral clustering. In *Introduction to spectral clustering*, pages 1 – 6.

J. A. Hartigan and M. A. Wong. 1979. Algorithm AS 136: A K-Means clustering algorithm. *Applied Statistics*, 28(1):100–108.

Anil K. Jain and Richard C. Dubes. 1988. *Algorithms for Clustering Data*. Prentice-Hall, Inc., USA.

Yaochu Jin. 2006. *Multi-objective machine learning*, volume 16. Springer Science & Business Media.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and paste based text summarization. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, NAACL 2000, page 178–185, USA. Association for Computational Linguistics.

Fabien Lauer and Christoph Schnörr. 2009. Spectral clustering of linear subspaces for motion segmentation. In *2009 IEEE 12th International Conference on Computer Vision*, pages 678–685.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Ulrike Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416.

Mihalcea Rada and Tarau Paul. 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.

Kamal Sarkar. 2008. Syntactic sentence compression: Facilitating web browsing on mobile devices. In *2008 International Conference on Information Technology*, pages 283–286.

Kamal Sarkar. 2009a. Centroid-based summarization of multiple documents. *TECHNIA–International Journal of Computing Science and Communication Technologies*, 2.

Kamal Sarkar. 2009b. Sentence clustering-based summarization of multiple text documents. *TECHNIA – International Journal of Computing Science and Communication Technologies*, 2:325–335.

Kamal Sarkar. 2010. Syntactic trimming of extracted sentences for improving extractive multi-document summarization. *Journal of Computing*, 2:177–184.

Kamal Sarkar. 2012a. An approach to summarizing bengali news documents. In *International Conference on Advances in Computing, Communications and Informatics*, ICACCI '12, page 857–862, New York, NY, USA. Association for Computing Machinery.

Kamal Sarkar. 2012b. Bengali text summarization by sentence extraction. *CoRR*.

Kamal Sarkar and Sivaji Bandyopadhyay. 2005. Generating headline summary from a document set. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'05, page 649–652, Berlin, Heidelberg. Springer-Verlag.

Oana Sidi, Oliver van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. Graph.*, 30(6):1–10.

J.C. Tilton. 1998. Image segmentation by region growing and spectral clustering with a natural convergence criterion. In *IGARSS '98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings. (Cat. No.98CH36174)*, volume 4, pages 1766–1768 vol.4.

X. Wan and J. Yang. 2008. Multi-document summarization using cluster-based link analysis. In *SIGIR-08*, page 299–306.

Xiaojun Wan and Jianwu Yang. 2006. Improved affinity graph based multi-document summarization. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 181–184, New York City, USA. Association for Computational Linguistics.