2018 21st International Conference on Computer and Information Technology (ICCIT), 21-23 December, 2018

Automatic Bangla Text Summarization Using Term Frequency and Semantic Similarity Approach

Avik Sarkar, Md. Sharif Hossen

Department of Information and Communication Technology
Comilla University, Comilla-3506, Bangladesh
ssavi.ict@gmail.com, mshossen@cou.ac.bd

***Abstract*— With the increasing amount of data within the cloud, it is harder to get the expected one. This leads to the idea of text summarization. Automatic text summarization is a tool for summarizing textual data into a short and concise piece of information via which people can have the idea about the content. Several approaches are introduced but there are a little amount of work has been done on Bangla text summarizing techniques due to some different and multifaceted structure of Bangla language. This paper illustrates the implementation of term frequency and semantic sentence similarity based summarizing approaches to summarize a single Bangla document. Removing stopwords, noisy words, lemmatization, tokenization has been done beforehand. Both of these methods return a bunch of top-ranked sentences to create a summary. The rank of a sentence is determined by the term frequency for the first approach and the sentence similarity for the second approach. The experimental result shows a favorable outcome for both of the approaches. Further improvements of these approaches certainly will return an enchanting outcome.**

Keywords—text mining, text summarization, python, nltk, wordnet, brown corpus, sentence similarity, word order similarity

I. INTRODUCTION

Data mining is a buzzword from several decades. It is the approach of finding new information by testing large pre-existing databases [1]. Text mining or text data mining is defined as the process of extracting a previously understandable, potential, unknown patterns or knowledge from the collection of massive and unstructured text data or corpus [2]. Text summarization is one of the challenging and important problems of text mining. Text summarization is the way to summarize single or multi-document texts and returns a short and concise amount of information that contains the gist of the whole content and gives a fair idea about the content. Therefore, processing documents are a perfunctory task, mostly due to the lack of standards [3]. Automatic text summarization is the task of generating a concise and fluent summary while storing main contents and overall theme [4].

Natural language processing is one of the most appealing research areas for data scientists and analyzers nowadays. In order to cope up with NLP research in the native languages, the researchers are working hard. In Bangladesh, researchers are also working in text mining sector on Bangla language. In this paper, we have discussed two mostly adapted extractive text summarization approaches to summarize a single document Bangla text.

A. Classification of Text Summarization

Automatic text summarization is such a process of generating summaries with the help of a gently written computer program. A text summarizing has three main steps. These steps are topic identification, making an interpretation, and the generation of summary [6]. Automatic text summarization can be classified based on several criteria. The different dimension of text summarization can be generally categorized based on its input type, purpose and output types [5]. There is also another type of classification which is based on the uses of external resources. The diagram below shows the summarization types under several criteria –

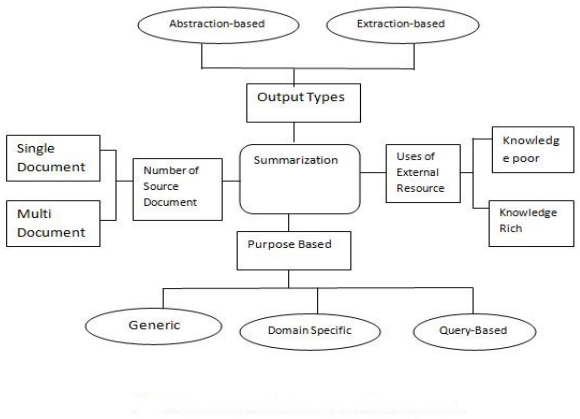

Fig. 1. Type of summarization

According to the number of documents, summarizations are of two types, namely, single and multi-document. According to the purpose, summarizations are three types, namely, generic, domain-specific and query oriented. According to the uses of external resources, summarizations are of three types, namely, knowledge-rich and knowledge poor. According to the types of output, summarizations are of two types, namely, abstraction based and extraction based.

This paper deals with single document extraction based summarization. Before getting into the detail, we discuss the abstraction and extraction based summarization techniques in a nutshell. Abstraction based summarization produces an abstract of the original text by using the interpretation procedure and generates a summary that expresses the same in a more concise way. Extractive summaries are produced by identifying important sentences which have directly been selected from the document. These types of summarizations are mostly found (Aliguliyev, 2009; Ko and Seo, 2008), where selected document sentences are coherently combined and

978-1-5386-9242-4/18/$31.00 ©2018 IEEE

compressed to exclude the sections of the sentences (Ganesan et al., 2010; Khan et al., 2015) [5]. In this approach, sentences are given some scores based on different criteria and then the sentences with relatively maximum rating are chosen to pick for summarization. It uses several types of natural language process (NLP) to retrieve information.

B. Usual Procedure of Text Summarization

Usually, the basic text summarization includes the following steps –

Step 1: The first step is to tokenize words or sentences, removing stopwords, stemming, lemmatization, word frequency calculation etc.

Step 2: The next step is associated with word scoring, sentence scoring, graph construction, semantic similarity calculation between sentences etc. These calculations have been done with the help of brown corpus and wordnet.

Step 3: The final step is to decide which sentences should be picked and how to place those sentences in order to produce a well understandable summary.

C. Usual Procedure for Extractive Summarization

Extractive summarization is such a summarization procedure in which the system returns the most relevant sentences as a summary without redundant appearances of sentences. These sentences are chosen based on compression rate, which is used to define the ratio between the length of the source and the summary text. Compression rate (about 5 - 30 %) is acceptable for a summary content. The procedures of the extractive technique include term frequency method, cluster-based method, sentence similarity-based method, fuzzy logic based method, neural network-based method, meta heuristic search approach, Query-based approach, topic-driven maximal marginal method, Concept obtained method, feature weight based regression analysis method, centroid-based summarization etc. The paper is designed as follows: In section II, we discuss the related works. Section III shows the methodology used in the research. Experimental analysis and discussion are mentioned in Section IV. Finally, Section V includes the conclusion with the future plan.

II. Related Works

In the last several years, a significant amount of extractive algorithms based on several features have been developed. R. Mihalcea and P. Tarau proposed a sentence ranking based approach in [7]. In this approach, the rank of the sentence will be decided by a similarity function. The approach in [7] is improved by Barrios, Lopez, Argerich, Wachenchauzer in [8]. They improved the sentence similarity calculation in three different ways and proposed three different techniques, namely, Longest Common Substring, Cosine TF-IDF, and BM25 / BM25+. A cue-based hub authority text summarization approach proposed by J.Zhang in [9] is used for multiple documents. It uses k-nearest neighboring to detect sub-topics. J. Steinberger and K. Jezek proposed a new evaluation measure in [10], which is based on the latent semantic analysis (LSA) and can capture the main topic of the document. Another technique proposed by Y. Onuyang in [11] is based on the hierarchical representation. To identify the subsumption of the words, he calculated point-wise minimum information (PMI) and later on higher PMI to determine the words are correlated or not. It was a multi-document summarization technique.

A query based extractive summarization technique proposed by J. Siddiki and K.Gupta in [12] is based on sentence clustering method. The semantic and syntactic similarity was the core idea to form the cluster. R. Nallapati, F. Zhai, B. Zhou proposed a recurrent neural network based sequence model for extractive summarization in [13]. The idea was based on maximizing the Rogue score of sentences with respect to gold summaries to select the sentences. They also proposed the novel training technique to train the system in an abstractive way to eliminate the need of approximate extractive labels generation. A Bayesian summarization model proposed by Daume et al in [14] was query-focused summarization technique. A Bayesian sentence-based topic model is proposed by Wang et al. [15] based on the term document and term sentence association. In [16], Celikyilmaz et al. proposed a two-phase hybrid model. The first one was the hierarchical topic model to discover the topic structure of all the sentences and then they compute the similarity between sentences based on human-provided summaries using a novel tree-based scoring technique. Using the scores in the second stage they train a regression model according to the lexical and structural characteristics of the sentences and use the model to form a new summarized document. Gong and Liu in [17] proposed a technique that calculates highly ranked sentences based on the semantic calculation that involves selecting the appearance of words in a sentence. They transformed the document into an N x N vector, where the column contains the words and the row contains the sentences. Later on, the scoring has been done by TF-IDF method. A feature-based sentence scoring approach is proposed by K. Meena and D. Gopalani in [19]. For calculating the score of a sentence, they used a function named fitness function. The fitness function works with about 21 different features. Each of the features has several scores and the sum of those features multiplied by a constant value will result in the sentence score. A hybrid function is introduced by AL-Khassawneh, Salim, Jarrah in [6]. This hybrid function is used as an improvised technique of triangle – graph based text summarization approach. This hybrid function consists of four different similarity measures to find similarity between sentences to create the graph. Several features are involved with that approach such as- title words, sentence lengths, sentence positions, numerical data, thematic words and sentence to sentence similarity. An approach to calculate the score of a sentence proposed by Ramanujam and Kaliappan uses the Naïve Bayesian Classifier based on timestamp strategy in [20]. This timestamp approach is used to achieve the coherent looking summary that extracts more relevant information from multiple documents. Scoring strategy is associated with word frequency, readability and comprehensibility. In Bengali language, K. Sarkar proposed a technique in [21] based on the sentence ranking associated with TF-IDF calculation of thematic term, positional values and the sentence length for an appearance in the document. Combining all the features together a sentence scoring equation is formed and k-most scored sentences are selected for the summary. Another approach based on TF-IDF and k-means clustering algorithm for sentence selection is proposed in [22] for Bangla text summarization. They applied TF-IDF approach for word scoring, and to select k-most scored sentence for generating the summary they use K-means clustering algorithm. A new semantic similarity measure proposed by Sinha, Jana, Dasgupta, and Basu in [23] is based on the hierarchical formation of words. The similarity will be the same if the words are in the same category. If they are in a different category, then the distance will be the similarity of the words. Based on this

concept they proposed a lexicon. Except for those approaches in Bengal language, there are more other similar approaches proposed by M.A. Uddin [24] and M. Imbrahim [25] based on TF-IDF.

III. Working Methodology

Here, we will use two different techniques to summarize Bangla text document. The first one is the term frequency and the other one is the semantic similarity-based approach. The idea for the first is picked from the concepts illustrated in [10]. The concept for the semantic similarity approach is picked from [18], which is basically a graph-based data model. For every approach, there are pre-processing we have to complete. Those are tokenization, removing stopwords, and lemmatization.

(a) Tokenization: There are basically two types of tokenization. First one is word tokenization and the other is sentence tokenization. In Bangla language, the period (.) is denoted by দাঁড়ি and the symbol by (।). Taking this issue in account a self-implemented tokenizing system is designed to tokenize both words and sentences.

(b) Removing Stopwords: Removing stopwords are basically those frequent words like আজ, কাল, অথবা, এবং, নতুবা, অন্যথায় etc., which actually do not have much effect on the meaning of a sentence. A self-implemented library of stopwords is designed in these cases although these are not enough and the contribution process is still going on.

(c) Lemmatization: There are several words in different formations like শহর, শহরে, শহরের, শহরতলী, শহরগুলো, শহরসমূহ etc. They basically denote the same meaning শহর but with the inclusion of terms like ‘s’, ‘es’ it shows the different form. To remove redundancy, the conversion of several formations into an actual word is also needed.

(d) Removing Duplicate Sentences: In a summary, duplicate line appearance is not expected. Hence, the removal of duplicate lines is applied.

A. Term Frequency Based Summarization

In this approach, word frequency will be calculated after the completion of pre-processing steps. The map of frequency is then filtered. That is, we have to ignore very high and very low-frequency terms. By ignoring those terms, we remove the noisy terms from the sentences. Noisy words are those terms that appear frequently or only a few times into the content but do not contain much information. By setting an apparent range of terms and maintaining the range, the system encounters only those terms that are relevant to the content. In this way, the sentences are ranked according to the frequency of the terms they contain. From those sentences, top K sentences are selected for the final summary. The algorithm of the whole procedure is given as follows:

TermFreqSum(input text T)

1. Tokenize sentences in T and save to S
2. Remove stopwords from sentences
3. For each sentence in S
4. Count frequency of each word *W* in a sentence
5. For each word W_i
6. M = Maximum(freq[W_i])
7. freq[W_i] = freq[W_i]/M
8. If(freq[W_i]>=max_cut or freq[W_i]<=min_cut)
9. Ignore Word W_i
10. For each i^{th} sentence in S
11. For each word in S
12. If *Wi* in freq
13. rank[i] += freq[*Wi*]
14. Top K sentences are selected for the final summary

In line 8, two terms, i.e., max_cut and min_cut are introduced. These variables contain two values, which can be put using observation between the ranges the summarizer will produce a decent result. According to our observation, we use min_cut = 0.1 and max_cut = 0.9. Applying this approach, a single document text summarization can be possible within a very short time. As there is a short task of only mapping the terms and counting them, as well as, prioritizing them by some other calculations, the execution time is much less. Due to using the max and min cut ranges, there are some possibilities to lose some information. While working with some random training data, it has been seen that it returns a decent summary. Redundant lines for summary will also be eliminated. This similar approach can also be applicable for multi-document summarization.

B. Semantic Similarity Based Summarization

In this approach, sentences are considered as a single node of a weighted graph after the pre-processing steps. Graph nodes are connected with one another where the weight of each edge is the similarity between two nodes, i.e., two sentences. To calculate the semantic, we have to consider two semantic similarity calculations, namely, word order and sentence order. The combination of those two similarities is actually the weight of the edge between any two sentences, i.e., the similarity between two sentences.

The similarity between a pair of words is calculated based on two functions- f(l) and f(h) where l is the shortest path between any two words in WordNet database and h is the height of their lowest common sub-summer (LCS). The WordNet is basically a database where words are stored based on the several categories. Synonym of a certain word is placed on the same category to get complete similarity results between them. The function f(l) and f(h) normalizes their values within 0 and 1. The equation is given below:

$$\tilde{s} = f(l)f(h) \dots\dots\dots (1)$$

where

$$f(l) = e^{\alpha l} \dots\dots\dots (2)$$

and

$$f(h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \dots\dots\dots (3)$$

Here, $\alpha \in [0, 1]$ and $\beta \in (0, 1]$.

Based on these calculations while choosing similar words, the algorithm chooses the most similar one. The calculation of the sentence semantic similarity is the combination of the similarity of both words, i.e., order similarity and semantic similarity. The semantic similarity is calculated using a cosine similarity between the semantic vectors of two sentences. The semantic vector is a vocabulary of a union of words of two sentences. If the word occurs in both of the sentences, then the semantic similarity is 1, otherwise, the similarity is calculated against all other words in the sentences. If the calculation leads to a threshold, φ, value, then the value of the word is φ else it is 0. The similarity value further can be attenuated by the information content hold by brown corpus. Equation is given below:

$$S_s = \frac{s_1 . s_2}{\|s_1\| . \|s_2\|} \dots\dots\dots\dots (4)$$

where $s_i = \tilde{s} . I(w_i) . I(\widetilde{w_i}) \dots\dots\dots\dots (5)$

$$I(w_i) = 1 - \frac{\log(n+1)}{\log(N+1)} \dots\dots\dots\dots (6)$$

n = number of times word w in corpus
N = number of words in the corpus

Word order similarity calculation can be done by calculating the word vector for each of the sentences and computing a normalized similarity. The word order vector will be the union of words in both of the sentences. If a word appears in both of the sentences, then the similarity will be recorded otherwise the similarity to the most similar words in the sentence is recorded if it does not cross the threshold value, η, else it is 0. The eqution is given below:

$$S_r = \frac{\|r_1 - r_2\|}{\|r_1 + r_2\|} \dots\dots\dots\dots (7)$$

where r_1 = word position vector for sentence 1

r_2 = word position vector for sentence 2

So, the final equation for the sentence similarity approach is as follows:

$$S(T_1, T_2) = \delta S_s + (1 - \delta) S_r$$

$$= \delta \frac{s_1 . s_2}{\|s_1\| . \|s_2\|} + (1 - \delta) \frac{\|r_1 - r_2\|}{\|r_1 + r_2\|} \dots\dots\dots (8)$$

The algorithm of the semantic similarity based approach is given as follows –

SemSimSum(Sentence s_1, Sentence s_2)

1. Set α = 0.2, β = 0.45, η = 0.4, φ = 0.2, δ = 0.85
2. Make a word vector by doing union of s_1 and s_2
3. *f(l)* is calculated using equation 2
4. *f(h)* is calculated equation 3
5. Word order similarity between s_1 and s_2 is calculated using equation 1
6. Semantic similarity is calculated using equation 4
7. Word order similarity is also calculated using equation 7
8. The sentence similarity is calculated equation 8
9. Finally, top ranked k sentence pairs are selected for the summary

As we previously remove the duplicate sentences, there is no chance of having any redundant sentences. This approach provides mostly the gist of the contents. Besides, it contains a more accurate result than the TF approach discussed above.

IV. Experimental Result and Analysis

To test the approaches described above we have used some training dataset randomly found. Dataset is collected from Facebook posts, online news content, and custom written texts from books. To compare those approaches with the human mind, we took help from some random users to make a summary. Given four samples with the input of different sizes and their corresponding output –

Sample 1: Size = 1.28 KB

Input: মানুষ সৃষ্টির শ্রেষ্ঠ প্রাণী । জগতের অন্যান্য প্রাণির সহিত মানুষের অয়ার্থক্য - মানুষ বিবেক ও বুদ্ধির অধিকারী । এই বিবেক, বুদ্ধি ও জ্ঞান নাই বলিয়া আর সকল প্রানী মানুষ অপেক্ষা নিকৃস্ট । জ্ঞান ও মনুষ্যত্বের উতকর্ষ সাধর করিয়া মানুষ জগতের বুকে অক্ষয়কীর্তি স্থাপন করিয়াছে, জগতের কল্যাণ সাধন করিতেছে, পশুবল ও অর্থবল মানুষকে বড় বা মহৎ করিতে পারে না । মানুষ বড় হয় জ্ঞান ও মনুষ্যত্বের বিকাশে । জ্ঞান ও মনুষ্যত্বের প্রকৃত বিকাশে জাতির জীবন উন্নত । প্রকৃত মানুষই জাটিম জীবনের প্রতিস্ঠা ও উন্নয়ন আনয়নে সক্ষম ।

Term Frequency Output: জ্ঞান ও মনুষ্যত্বের উতকর্ষ সাধর করিয়া মানুষ জগতের বুকে অক্ষয়কীর্তি স্থাপন করিয়াছে , জগতের কল্যাণ সাধন করিতেছে , পশুবল ও অর্থবল মানুষকে বড় বা মহৎ করিতে পারে না ।

Semantic Sentence Similarity Output: মানুষ বড় হয় জ্ঞান ও মনুষ্যত্বের বিকাশে ।

A Random User Output: সৃষ্টির শ্রেষ্ঠ প্রানী হিসেবে জ্ঞান ও মনুষ্যত্বের গুনে মানুষ জগতে যে অমরকীর্তি গড়ে তুলেছে পশুবল ও অর্থবল দিয়ে তা কখনো সম্ভব নয় ।

Using the input as 1.28 KB, the term frequency takes 0.058 seconds to produce the output while graph theoretic needs 10 seconds.

Sample 2: Size = 1.81 KB

Input: ভাত বাঙালির বহুকালের প্রিয় খাদ্য । সরু সাদা চালের গরম ভাতের কদর সবচাইতে বেশি ছিল বলে মনে হয় । পুরোনো সাহিত্যে ভালো খাবারের নমুনা হিসেবে যে-তালিকা দেওয়া হয়েছে, তা হলো কলার পাতায় গরম ভাত, গাওয়া ঘি, নালিতা শাক, মৌরালা মাছ আর খানিকটা দুধ । লাউ, বেগুন ইত্যাদি তরকারি প্রচুর খেত সেকালের বাঙালিরা, কিন্তু ডাল তখনো বোধহয় খেতে শুরু করেনি । মাছ তো প্রিয় বস্তুই ছিল । বিশেষ করে ইলিশ মাছ । শুঁটকির চল সেকালেও ছিল বিশেষ করে দক্ষিণাঞ্চলে । ছাগলের মাংস সবাই খেত । হরিণের মাংস বিয়েবাড়িতে বা এরকম উৎসবে দেখা যেত । পাখির মাংসও তা-ই । সমাজের কিছু লোক শামুক খেত । ক্ষীর, দই, পায়েস, ছানা-এসব ছিল বাঙালির নিত্যপ্রিয় । আম-কাঁঠাল, তাল-নারকেল ছিল প্রিয় ফল । খুব চল ছিল নাড়ু, পিঠেপুলি, বাতাসা, কদমা-এসবের। মসলা-দেওয়া পান পান খেতে সকলে ভালবাসত ।

Term Frequency Output: খুব চল ছিল নাড়ু, পিঠেপুলি, বাতাসা, কদমা-এসবের। মসলা-দেওয়া পান পান খেতে সকলে ভালবাসত । মাছ তো প্রিয় বস্তুই ছিল । ভাত বাঙালির বহুকালের প্রিয় খাদ্য । পুরোনো সাহিত্যে ভালো খাবারের নমুনা হিসেবে যে-তালিকা দেওয়া হয়েছে, তা হলো কলার পাতায় গরম ভাত, গাওয়া ঘি, নালিতা শাক, মৌরালা মাছ আর খানিকটা দুধ ।

Semantic Sentence Similarity Output: মাছ তো প্রিয় বস্তুই ছিল । আম-কাঁঠাল, তাল-নারকেল ছিল প্রিয় ফল । বিশেষ করে ইলিশ মাছ । শুঁটকির চল সেকালেও ছিল বিশেষ করে দক্ষিণাঞ্চলে ।

A Random User Output: বাঙালি জাতির জীবনযাত্রার পরিচয়ের মধ্যে খদ্যাভ্যাস অন্যতম। প্রাচীনকাল থেকে এদেশের মানুষ বিচিত্র ধরনের সাধারণ খাবার খেত। উৎসব বা বিয়েতে হরিণের মাংস পরিবেশন করা হতো। সমাজের সকল স্তরের ও অঞ্চলের খাদ্যাভ্যাস প্রায় একই ধরনের ছিল।

Using the input as 1.81 KB, the term frequency takes 0.058 seconds to produce the output while graph theoretic needs 13 seconds.

Sample 3: Size = 3.65 KB

Input: শিক্ষা বা জ্ঞান অর্জন হলো সাধনার ব্যাপার । তবে এই সাধনার সাধক হতে হবে শিষ্যের নিজেকেই । একজনের সাধনা কখনও অন্য কেউ করে দিতে পারে না। যার সাধনা তাকেই সাধন করতে হয় । অন্যথায় সাধনার ফলাফল কখনই আশানুরূপ হয় না। আমাদের অনেকের মধ্যেই একটি বিশেষ প্রবণতা লক্ষ্য করা যায়, তা হলো গুরু বা শিক্ষকের উপর সম্পূর্ণ ভরসা করে বসে থাকা । আমাদের এই প্রবণতার কারণেই আমাদের শিক্ষা শতভাগ পরিপূর্ণ হয় না । গুরু কিংবা শিক্ষক নিঃসন্দেহে একজন ছাত্রের নিকট ভরসার পাত্র হবে এটাই স্বাভাবিক । কিন্তু তার মানে এই নয় যে, গুরুই তার শিক্ষাকে অন্তরে গেঁথে দেবেন। অন্তরে গেঁথে দেওয়ার দায়িত্ব গুরুর নয় । গেঁথে নেওয়ার দায়িত্ব শিষ্যের । গুরু বড়জোর পথ দেখিয়ে দিতে পারেন । গুরু শুধুমাত্র শিষ্যকে বলে দিতে পারেন কোন পথ তার জন্য উত্তম, কোন পথে, কিভাবে হেটে গেলে সে তার কাঙ্ক্ষিত বস্তুর দেখা পেতে পারে । কিন্তু এরপরের সব দায়িত্বই শিষ্যের । গুরুর দেখানো পথে, গুরুর নির্দেশিত পন্থায় হেটে যেতে হবে শিষ্যের নিজেকেই । সঠিকভাবে সে পথ পাড়ি দিয়ে কাঙ্ক্ষিত বস্তুটি অর্জন করে আনা শিষ্যেরই দায়িত্ব । আমরা প্রায়ই ছাত্রের খারাপ ফলাফলের জন্যই শিক্ষককেই দোষারোপ করি । কিন্তু খারাপ ফলাফলের জন্য কখনও শিক্ষক দায়ী নয়, বরং ছাত্ররাই দায়ী । কিন্তু শিক্ষকের দেখানো পথে ছাত্র যদি হাঁটতে না পারে সে অযোগ্যতা

শুধুমাত্র ছাত্রের । শিষ্য পথভ্রষ্ট হলে সে ত্রুটির ভার **শিষ্যকেই** বহন করতে হয় । সে ভার গুরুর উপর চাপিয়ে দিলে তা কখনও সুবিচার হয় না । গুরু শিষ্যের মঙ্গল কামনা করেন এবং কল্যাণের পথই দেখিয়ে থাকেন । **কিন্তু** সেই কল্যাণ সাধনে যদি শিষ্যের সাধনায় ত্রুটি থাকে, তবে তা একান্তই **শিষ্যের** অপারগতা ।

Term Frequency Output: গুরুর দেখানো পথে, গুরুর নির্দেশিত পন্থায় হেটে যেতে হবে শিষ্যের নিজেকেই । গুরু শুধুমাত্র **শিষ্যকে** বলে দিতে পারেন কোন পথ তার জন্য উত্তম, কোন পথে, কিভাবে **হেটে** গেলে সে তার কাঙ্ক্ষিত বস্তুর দেখা পেতে পারে ।

Semantic Sentence Similarity Output: **গুরু** বড়জোর পথ দেখিয়ে দিতে পারেন । গুরু শুধুমাত্র শিষ্যকে বলে দিতে পারেন কোন পথ তার জন্য উত্তম, কোন পথে, কিভাবে হেটে গেলে সে তার কাঙ্ক্ষিত বস্তুর দেখা পেতে পারে ।

A Random User Output: বিদ্যার সাধনা **শিষ্যকে** নিজে অর্জন করতে হয়, গুরু উত্তরসাধক মাত্র । গুরু কেবল উত্তম পথ **দর্শ** করান । কিন্তু সেপথে সাধনে করে সিদ্ধি লাভ করতে শিষ্যকেই ।

Using the input as 3.65 KB, the term frequency takes 0.066 seconds to produce the output while graph theoretic needs 22 seconds.

Sample 4: Size = 10.9 KB

We have used another sample with an input size of 10.9 KB where frequency and graph-theoretic approaches take 0.455 and 100 seconds respectively to produce their outputs. For page limitation, we cannot include it. We have uploaded it on the internet [26].

Figure 2 shows the runtime complexity (measured in seconds) with respect to the text size (measured in KB).

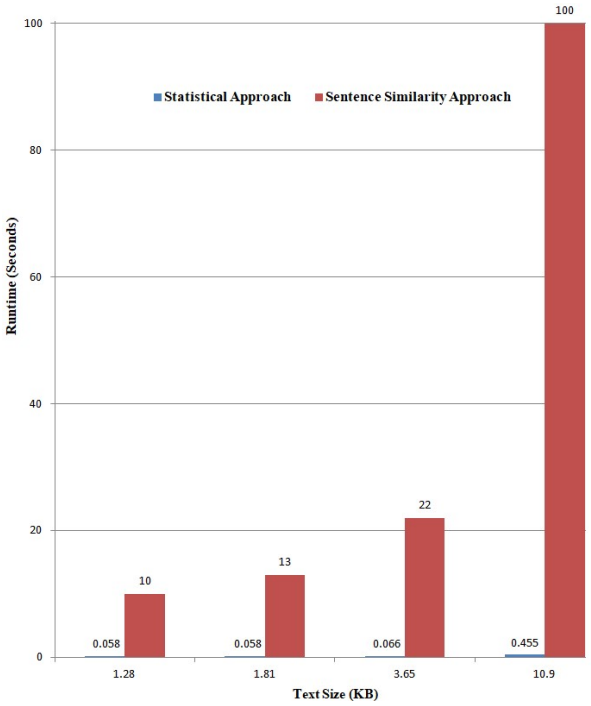

Fig. 2. Runtime vs. text size

Applying term frequency method within this text document results a list of noisy words like অর্জন, হেটে, স্বাভাবিক, শিক্ষাকে etc. Based on the max_cut and min_cut ranges, this method removes most frequent and less frequent terms in order to retrieve the best possible summary. In semantic similarity approach, unfortunately, we cannot manage any valid Bangla WordNet to perfectly classify the distance. The synonyms are here considered as different words. The distance also cannot perfectly be measured due to the lack of WordNet. Brown corpus does not contain any Bangla resources and hence the calculations are incomplete. Improvement of all of these issues will certainly provide a far better result than the current result. Besides, these approaches can be improved by extracting several other features like header terms, cue words, title similarity, thematic features, named entity etc. Based on the execution time, the term frequency approach is pretty faster (shown in Figure 2) but for retrieving a better quality summary, the semantic similarity approach is much better with average accuracy of 69.13% .

V. Conclusion and Future Works

In this paper, to summarize a single document Bangla text, two approaches, namely, term frequency and semantic sentence similarity-based approaches are implemented. The first one is based on the calculations of the frequency of terms in the content and the other one is based on the semantic and word order similarity between sentences. Both approaches are extraction based and return a set of most relevant sentences from which a certain number of sentences are selected to produce a summary. Similarity measurement is a great issue in text summarization problem. So, the improvement of the quality of similarity measurement by adding several features, e.g., enriching WordNet for Bangla language and contributing Bangla text resources in brown corpus could certainly be a fruitful choice. From the analysis result, we see that frequency summarizer is pretty faster than but for retrieving a better quality summary, the semantic similarity approach is much better.

As a future work to improve the quality of sentence similarity approach, we would like to continue our study to develop a category based Bangla WordNet where words will be arranged under different categories. Base words and synonyms will also be arranged within the same category.

REFERENCES

[1] Verloren, “Wikipedia Data Mining Article,” 2002.

[2] Yu Zhang, Mengdong Chen, and Lianzhong Liu, “A review on Text Mining,” IEEE International Conference on Software Engineering and Service Science, Beijing, China, 2015.

[3] Juan-Manuel, and Torres-Moreno, “Automatic Text Summarization (Cognitive Science and Knowledge Management),” 1st Edition, Wiley-ISTE, 2014.

[4] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut, “Text Summarization Techniques: A Brief Survey,” arXiv, USA, 2017.

[5] Yogan Jaya Kumar, Ong Sing Goh, Halizah Basiron, Ngo Hea Choon, and Puspalata C Suppiah, “A Review on Automatic Text Summarization Approaches,” Journal of Computer Science, vol. 12, Iss. 4, pp. 178-190, 2016.

[6] Yazan Alaya AL-Khassawneh, Naomie Salim, and Mutasem Jarrah, “Improving Triangle-Graph Based Text Summarization using Hybrid Similarity Function,” Indian Journal of Science & Technolog, vol. 10, Iss. 8, 2017.

[7] R. Mihalcea, and P. Tarau, “Textrank: Bringing Order into Texts,” Proceedings of EMNLP, Association for Computational Linguistics, Barcelona, Spain, pp. 404-411, 2004.

[8] F. Barrios, F. Lopez, L. Argerich, and R. Wachenchauzer, “Variations of the Similarity Function of TextRank for Automated Summarization,” Argentine Symposium on Artificial Intelligence, pp. 65-72, 2015.

[9] J. Zhang, L. Sun, and Q. Zhou, “Cue-based Hub-Authority approach for Multi-document Text Summarization,” IEEE International Conference on Natural Language Processing and Knowledge Engineering, pp. 642 –645, China, 2005.

[10] Josef Steinberger, and Karel Jezek, “Evaluation Mesaures for Text Summarization,” Computing and Informatics, vol. 28, pp. 1001–1026, 2009.

[11] Y. Ouyang, and W. Li, Q. Lu, “An Integrated Multi-document Summarization Approach based on Word Hierarchical Representation”, Proceedings of the ACL-IJCNLP, pp. 113-116, China, 2009.

[12] T. J. Siddiki, and V. K. Gupta, “Multi-document Summarization using Sentence Clustering”, IEEE Proceedings of International Conference on Intelligent Human Computer Interaction, India, 2012.

[13] Ramesh Nallapati, Feifei Zhai, Bowen Zhou, and SummaRuNNer, “A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents,” Thirty-First AAAI Conference on Artificial Intelligence, 2016.

[14] Hal Daumé III, and Daniel Marcu, “Bayesian Query-focused Summarization,” Proceedings of the International Conference on Computational Linguistics, Association for Computational Linguistics, pp. 305–312, 2009.

[15] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong, “Multi-Document Summarization Using Sentence-Based Topic Models,” Proceedings of the ACL-IJCNLP, Association for Computational Linguistics, pp. 297–300.

[16] Asli Celikyilmaz, and Dilek Hakkani-Tur, “A Hybrid Hierarchical Model for Multi-Document Summarization,” Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp. 815-824, 2010..

[17] Yihong Gong, and Xin Liu, “Generic Text Summarization Using Relevance Measure, and Latent Semantic Analysis,” In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, USA, pp. 19–25, 2001.

[18] Yuhua Li, McLean, Zuhair A. Bandar, James D. O'Shea, and Keely Crockett, “Setnece Similarity Based on Semantic Nets and Corpus Statistics,” IEEE Transactions on Knowledge and Data Engineering, vol. 18, Iss. 8, pp. 1138 – 1150, 2006.

[19] Yogesh Kumar Meena, and Dinesh Gopalani, “Evolutionary Algorithms for Extractive Automatic Text Summarization,” Procedia Computer Science, International Conference on Intelligent Computing, Communication & Convergence, Interscience Institute of Management and Technology, Bhubaneswar, Odisha, India, pp. 244-249, 2015.

[20] Nedunchelian Ramanujam, and Manivannan Kaliappan, “An Automatic Multi document Text Summarization Approach Based on Naive Bayesian Clasifier Using Timestamp Strategy,” The Scientific World Journal, Hindawi Publishing Corporation, 2016.

[21] Kamal Sarkar, “Bengali Text Summarization by Sentence Extraction,” Proceedings of International Conference on Business and Information Management,” NIT Durgapur, PP 233-245, 2012.

[22] Sumya Akter, Arsy Siddika Asa, Md. Palash Uddin, Md. Delowar Hossain, Shikhor Kumer Roy, and Masud Ibn Afjal, “An Extractive Text Summarization Technique for Bengali Document(s) using K-means Clustering Algorithm,” IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Dhaka, Bangladesh, 2017.

[23] Mnajira Sinha, and Abhik Jana, Tirthankar Dasgupta, and Anupam Basu, “A New Semantic Lexicon and Similarity Measjure in Bangla,” Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon, Mumbai, India.

[24] M. A. Uddin, K. Z. Sultana, and M. A. Alom, “A Multi-Document Text Summarization for Bengali Text,” IEEE International Forum on Strategic Technology (IFOST), Bangladesh, 2014.

[25] M. I. A. Efat, M. Ibrahim, and H. Kayesh, “Automated Bangla Text Summarization by Sentence Scoring and Ranking,” IEEE International Conference on Informatics, Electronics & Vision (ICIEV), Bangladesh, 2013.

[26] https://github.com/sharifhossen/Bangla-Text-Summarization-using-Graph-Theoretic-and-Frequcny-Summarizer, 2018.