NEURAL APPROACH IN EXTRACTIVE BANGLA TEXT SUMMARIZATION

Course Code: SE 801 Project

Final Report

Submitted by
Md. Sefat-E-Mahadi
BSSE 0839

Supervised by
Dr. Ahmedul Kabir
Assistant Professor
Institute of Information Technology
University of Dhaka

Submitted to
SPL-3 Coordinators
Institute of Information Technology
University of Dhaka

LETTER OF TRANSMITTAL

17 December, 2019
Coordinator
Software Project Lab 3
Institute of Information Technology
University of Dhaka
Subject: Submission of Software Requirement Specification report on ”Neural Approach in Extractive Bangla Text Summarization”.

Sir,
I, Md. Sefat-E-Mahadi, am submitting my Software Requirement Specification report with due respect. Despite my best effort, it might still contain some errors. I hope that you would be kind enough to accept this report.

Yours sincerely,
Md. Sefat-E-Mahadi
BSSE 0839

LETTER OF ENDORSEMENT

The report SRS by Md. Sefat-E-Mahadi has been submitted to me prior to final submission. I have gone through all the contents of the report and found that the information provided here is valid and not exaggerated. I hereby gladly assert the validity of the report.

……..

Dr. Ahmedul Kabir
Assistant Professor
Institute of Information Technology
University of Dhaka

Table of Content

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

This document contains functional, non functional and support requirements and establishes a requirements baseline for the development of the system. This document contains requirements, description, use cases, activity diagrams of “Neural Approach in Bangla Extractive Text Summarization”. It also contains context based, component based and interface design. This document will help understanding the project as functionalities and implementation details of the project are clearly mentioned here. The information about the requirements here have been organized systematically so that everyone can easily figure out a summarized concept about the project. It will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Objectives and Scopes

Extractive summarization means identifying important sections of the text and generating to produce a subset of the sentences from the original text. The extraction is made according to the defined metric without making any changes to the texts. It is a hot topic in natural language processing. Several techniques have been developed since the beginning of natural language processing such as sentence ranking [Rama & Bhargav 2017], graph based method [Gunes 2000], deep learning approach [Akash & Avishek 2017; Aditya & Divij 2018] etc. Although almost all of them have been tried for bangla text summarization [Kamal 2012; Efat, Ibrahim, & Humayun 2013; Majharul, Surayia & Zerina 2017] but the deep learning approach is one the least tried. So in this project I will try to implement deep learning neural network approach to convert bangla text document to its summary.

1.3 Challenges

Implementing a text summarization for the language Bengali was not as straightforward as it is for the more global language English. Numerous summarization projects has been carried out on English, and this lead to the availability of easily accessible packages and libraries which conducts the preprocessing of the test data in seconds. For Bengali, no such library could be found, hence codes had to be written from scratch to make the system a success.

Chapter 2

Literature Review

Erkan and Radev [2000] proposed " LexRank: Graph-based Lexical Centrality as Salience in Text Summarization " an improved version of TextRank where the sentence similarity is calculated as ratio of multiplication of term frequency and inverse domain frequency of common words and union of all words of both sentences. This approach is applicable not only for single document summarization but also multi-document summarization as well.

Mihalcea and Paul [2004] proposed a graph based sentence extraction approach titled as "TextRank: Bringing Order into Texts" to generate summary that constructs a graph with each sentence as vertex and each weight edge represents cohesion among a pair of sentences. Then sentences with highest edge are selected as summary.

Kaikhah [2004] proposed a neural network approach to extract sentences. Paragraph follows title, Paragraph location in document, Sentence location in paragraph, First sentence in paragraph, Sentence length, Number of thematic words in the sentence, Number of title words in the sentence are considered as features. A supervised dataset was used to train a feedforward network to generate a model which was used to generate summary for unseen documents.

Kamal [2012] presented keyphrases extraction based Bangla and English text summarization which is a variant of an existing method [25]. The algorithm for sentence selection and summary generation works in two phases. Phase-1 uses sentence position and document's keyphrases. If phase-1 fails to generate the summary of user desired length, phase-2 is activated and select more sentences.

Efat, Ibrahim, & Humayun [2013] introduced a method for Banglatext summarization by sentence scoring and ranking. Their system has three segments: (i) pre-processing the test document, (ii) sentence scoring, and (iii) generating summary. Sentence scoring is depended on term frequency, position, cue words and skeleton of the document. Here, skeleton of the document consists of the words in the title and headers. It is noticeable that evaluation has been accomplished using only 10 documents and standard evaluation has not been turned here to calculate precision, recall and F-measure.

Aditya, Divij and Manish [2017] introduced weighted sentence embedding approach for sentence extraction. Words were converted to fixed length numeric vectors and sentence vector was computed by averaging each vector of word of a sentence. Several other features also collected such as sentence length, sentence to sentence cohesion, sentence position, mean term frequency and inverse domain frequency, occurrences of proper nouns, degree of centrality.

Jain, Bhatia, Manish [2017] proposed “Extractive Text Summarization using Word Vector Embedding” that embeds each word using a pre-trained GLOVE vector, takes other sentence features such as sentence-to-sentence cohesion, sentence-to-centroid cohesion, depth of tree, keyword similarity, occurance of proper nouns to train a feedforward network.

Haque, Pervin and Begum [2017] proposed “An Innovative Approach of Bangla Text Summarization by Introducing Pronoun Replacement and Improved Sentence Ranking ” where each sentence were ranked based on number of proper nouns nouns, summation of sentence similarities, and numerical data.

Akash, Avishek and Akshay [2018] introduced word embedding and sentence embedding for sentence extraction. Words were represented as numeric vector of fixed dimension. Sentence vector was computed using word vector and facebook fasttext library. Then a neural model was trained to predict the probability of a sentence to be included into summary. Pagination techniques was used to pick up the whole concept of a document.

Chapter 3

Quality Function Deployment

Quality Function Deployment (QFD) is a quality management technique that translates the needs of the customer into technical requirements for software. QFD's main aim is understanding that what is valuable to the customer and then deploys these values throughout the engineering process. The requirement specifications are provided in below sections.

Normal Requirements: Normal requirements include the objectives and goals that are stated during meeting with a customer for a product or system. We found some such objectives and goals during requirement analysis in inception step:

- Text document will be summarized in extractive approach that depicts the main theme and content of the original document.
- User friendly interface.

Exciting Requirements:

- Feature to download summary
- Summary highlighting in main article.

Chapter 4

Scenario Based Modeling

Scenario based modelling is the first phase where the usage of product can be visualized. This model enables us to get a vivid idea how user will use the product. In the following, we describe how the user story and use case.

4.1 Scenario

User will download the installer and after installation of the installer a graphical user interface will pop up with a text box. The user will paste original document into it and provide ratio of the summary and original document by sliding a slide bar, after clicking the ‘summarize’ button an extractive summary will be generated inside a text field beside the input text box. Summary generation steps are as follows, preprocessing, word embedding by using Word2Vec model, sentence embedding using encoder-decoder sequence model, collections of other features such as length, positional value, degree of centrality, number of cue words, similarity to tittle. Then a final neural network model will be generated to predict probabilities of sentences to be included or not included in the summary.

4.2 Use Case Diagrams

After analyzing the user story we found three actors who will directly use the system as a system operator. Primary actors are those who will play action and get a reply from the system whereas secondary actors only produce or consume information.

Following are the actors of Bangla Text SUmmarization–
1. Normal User
2. System

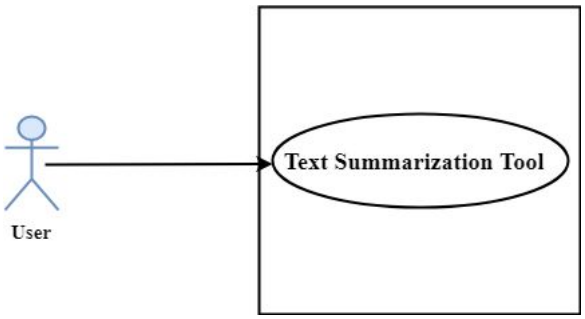


Figure 1: Use case 0

Primary actor: User
Goal in context: The diagram refers to the overview of the summarization tool.

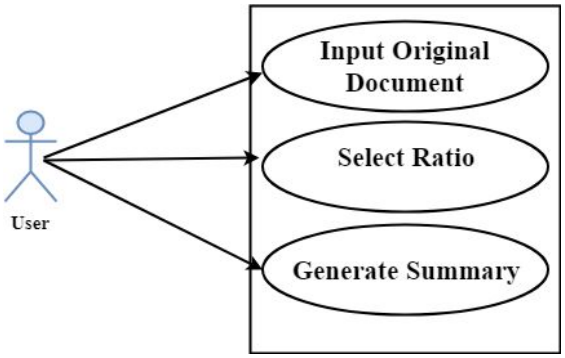


Figure 2: Use Case 1

Primary User: User
Goal in the context: This diagram refers to the details of text summarization tool

4.3 Activity Diagram

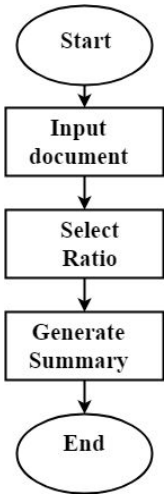


Figure 3: Activity diagram of use case 1

4.3 Swimlane Diagram

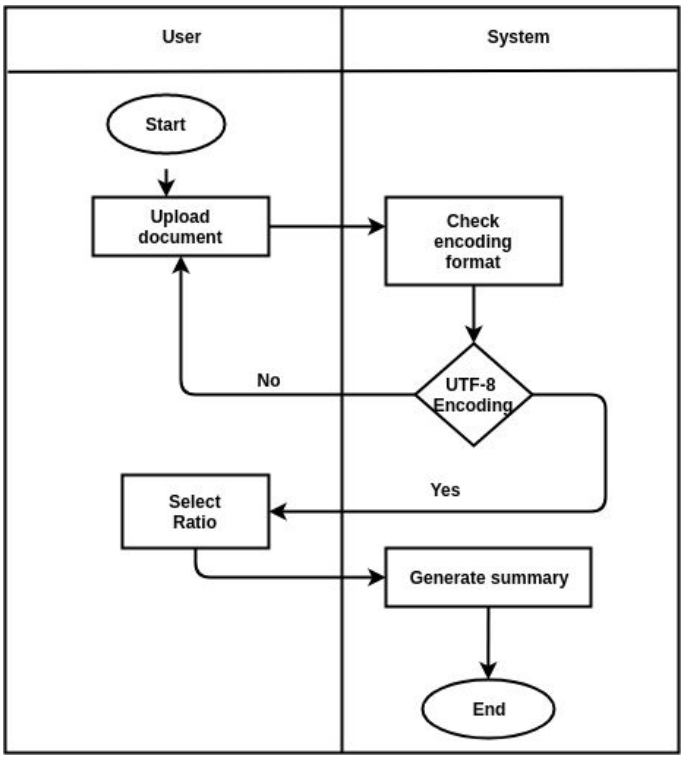


Figure 4: Swimlane diagram of use case 1

Chapter 5
Methodology

Whole summarization process can be divided as follows:

1. Preprocessing
2. Word embedding
3. Sentence embedding
4. Features collection
5. Model training and summary generation

5.1 Preprocessing

Preprocessing contains following steps,

1. Loading entire training data set.

2. Tokenization of each documents to word level .

3. Removing stop words and unnecessary characters from tokenized word list. Stops words in Bangla such as ‘ও’, ‘আর’, ‘এবং’, ‘এ’ are deleted.

4. Stemming each word to its root level. Bangla words such as ‘রক্ষার’, ‘পাকিস্তানের’, ‘স্বাধীনতার’, ‘লড়াইকে’ are converted to their root word such as 'রক্ষা', 'পাকিস্তান', 'স্বাধীনতা', 'লড়াই' . Word stemming decreases vocabulary size and increases term frequency of a word.

5. Pairing each word with theirs corresponding neighbour words according to a fixed window size.

6. Use one hot encoding to convert each word to a unique array.

Example:

Consider the sentence below,

“অখণ্ড পাকিস্তান রক্ষার অজুহাতে ঢাকা আজ ধ্বংসপ্রাপ্ত ও ভয়ের নগরী”

5.1.1 Tokenization

Tokenization is the process of splitting each sentence into separate words. In order to check for occurrences of words in the sentences and to increment the scores for any positive matches, tokenization had to be done.

After tokenization the sentence will be converted to,

[’অখণ্ড’ , ’পাকিস্তান’ , ’রক্ষার’ , ’অজুহাতে’ , ’ঢাকা’ , ‘আজ’ , ‘ধ্বংসপ্রাপ্ত’ , ’ও’ , ‘ভয়ের’ , ’নগরী’]

5.1.2 Stopword Removal

Words that hold any contribution for expressing the meaning of a sentence and have very little meaning themselves are called stop words. Bengali sentences are often filled with numerous stopwords. Bengali language and its grammar are designed as such that one has to use stopword to make it complete. Words such as ‘অবশ্য’, ‘এই’, ‘অতএব’,’অথচ’,’অনুযায়ী’ are few of words from the enormous list of stopwords. All such words are detected and are removed before the scoring starts. If the stopwords are not removed then they tend to take up a lot of computational resources and as these words are likely to be repeated, they appear to be scored higher than the actual meaningful words and eventually contribute to generating inaccurate summaries.

5.1.3 Stemming

In Bengali, a certain root word can be manipulated in multiple ways to make it best suited with the sentence and the context it is used for. For example, the word ‘কাজ’ can be used as ‘কাজটি’ etc, but all of these words originate from the same root word which is ‘কাজ’, hence to make the system’s scoring mechanism more accurate and relevant, a stemming mechanism is incorporated in preprocessing, which simply converts all the words to their very root version. If the following words are taken as an example, ‘কাজটি’ etc will all be converted to ‘কাজ’. So that each time the words come up, the system’s scoring mechanism will recognise them and treat the words as the same word as the root word. A rule based generic Bengali stemmer as implemented in [Mahmud, Arfin, Razzaque, 2014] has been used which converts a Bengali word into its stemmed form.

After stemming each word the sentence will be converted to

[‘অখণ্ড’, ‘পাকিস্তান’, ‘রক্ষা’, ‘অজুহাত’, ‘ঢাকা’, ‘আজ’, ‘ধ্বংস’, ‘প্রাপ্ত’, ‘ভয়’, ‘নগরী’]

5.1.4 Word Pairing

Now a neural network will be trained by feeding it word pairs found in training documents to generate word embedding. The below example shows some of the training samples (word pairs) which would be taken from the sentence “অখণ্ড পাকিস্তান রক্ষা অজুহাত ঢাকা আজ ধ্বংস প্রাপ্ত নগরী”.

Assuming window length is 2, here bold words are representing a window.

Window position	Pairs
অখণ্ড পাকিস্তান রক্ষা অজুহাত ঢাকা আজ ধ্বংস প্রাপ্ত নগরী	(অখণ্ড,পাকিস্তান) (অখণ্ড, রক্ষা)
অখণ্ড পাকিস্তান রক্ষা অজুহাত ঢাকা আজ ধ্বংস প্রাপ্ত নগরী	**(**পাকিস্তান, অখণ্ড**)** **(**পাকিস্তান, রক্ষা**)** **(**পাকিস্তান, অজুহাত**)**
অখণ্ড পাকিস্তান রক্ষা অজুহাত ঢাকা আজ ধ্বংস প্রাপ্ত ভয় নগরী	**(**রক্ষা, অখণ্ড**)** **(**রক্ষা,পাকিস্তান**)** **(** রক্ষা,অজুহাত**)** **(**রক্ষা,ঢাকা**)**
অখণ্ড **পাকিস্তান রক্ষা অজুহাত ঢাকা আজ** ধ্বংস প্রাপ্ত ভয় নগরী	(অজুহাত,পাকিস্তান) (অজুহাত,রক্ষা) (অজুহাত,ঢাকা) (অজুহাত,আজ)
অখণ্ড পাকিস্তান **রক্ষা অজুহাত ঢাকা আজ ধ্বংস** প্রাপ্ত ভয় নগরী	(ঢাকা,রক্ষা) (ঢাকা,অজুহাত) (ঢাকা, আজ) (ঢাকা,ধ্বংস)
অখণ্ড পাকিস্তান রক্ষা **অজুহাত ঢাকা আজ ধ্বংস প্রাপ্ত** ভয় নগরী	(আজ,অজুহাত**)** **(**আজ,ঢাকা**)** **(**আজ,ধ্বংস **)** **(**আজ,প্রাপ্ত)

অখণ্ড পাকিস্তান রক্ষা অজুহাত **ঢাকা আজ ধ্বংস প্রাপ্ত ভয়** নগরী	(ধ্বংস,ঢাকা) (ধ্বংস,আজ) (ধ্বংস,প্রাপ্ত) (ধ্বংস, ভয়)
অখণ্ড পাকিস্তান রক্ষা অজুহাত ঢাকা **আজ ধ্বংস প্রাপ্ত ভয় নগরী**	(প্রাপ্ত,আজ) (প্রাপ্ত,ধ্বংস) (প্রাপ্ত,ভয়) (প্রাপ্ত,নগরী)
অখণ্ড পাকিস্তান রক্ষা অজুহাত ঢাকা আজ **ধ্বংস প্রাপ্ত ভয় নগরী**	(ভয়,ধ্বংস) (ভয়,প্রাপ্ত)

Table 1: Pairing of words in a sentence

5.1.5 One Hot Encoding

It is not possible to feed words directly to a neural network, so after completing pairing, one hot encoding technique will be applied to encode each word to numerical form. Table 2 depicts an example of one hot encoding technique. Vocabulary is generated from training dataset.

Word	**Encoding vector**
অখণ্ড	[**1**, 0, 0, 0, 0, 0, 0, 0, 0]
পাকিস্তান	[0, **1**, 0, 0, 0, 0, 0, 0, 0]
রক্ষা	[0, 0, **1**, 0, 0, 0, 0, 0, 0]
অজুহাত	[0, 0, 0, **1**, 0, 0, 0, 0, 0]
ঢাকা	[0, 0, 0, 0, **1**, 0, 0, 0, 0]
আজ	[0, 0, 0, 0, 0, **1**, 0, 0, 0]
ধ্বংস	[0, 0, 0, 0, 0, 0, **1**, 0, 0]
প্রাপ্ত	[0, 0, 0, 0, 0, 0, 0, **1**, 0]
নগরী	[0, 0, 0, 0, 0, 0, 0, 0, **1**]

Table 2: One hot encoding of words.

5.2 Word Embedding

According to Mikolov [2013] numerical representation of words can be achieved by training a shallow neural network. The network will be trained to where each input will be one hot encoding vector of a word and output will be paired word that was paired during preprocess step. After training the network weights of first layer will be combined to embed the output word.

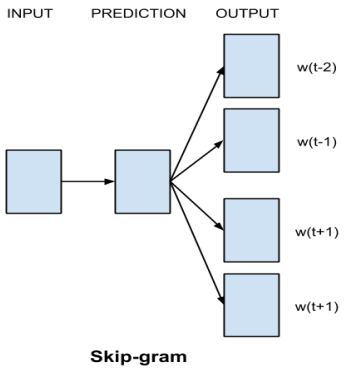


Figure 5: Skip-gram model of word embedding.

For example, word pairs generated in the previous step along with their corresponding encoding vector can be used to train a skip gram word embedding model. Encoding vector of the first word of the pair will be input and other vector will be output.

Word pair	Input	Output
(অখণ্ড,পাকিস্তান)	[**1**, 0, 0, 0, 0, 0, 0, 0, 0]	[0, **1**, 0, 0, 0, 0, 0, 0, 0]
(অখণ্ড, রক্ষা)	[**1**, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, **1**, 0, 0, 0, 0, 0, 0]
(পাকিস্তান, অখণ্ড**)**	[0, **1**, 0, 0, 0, 0, 0, 0, 0]	[**1**, 0, 0, 0, 0, 0, 0, 0, 0]

Table 3: Input output pair of word2vec model

The next represents example of word embedding with window length = 3, learning rate =0.01, epoch =5000, min_count =1 and embedding_dimension =10

Word	Embedding vector
অখণ্ড	[0.3601206, 0.42496127, 0.29524784, 0.02665084, 0.30171084, 0.15456992, 0.68505177, 0.36933134, 0.69138272, 0.17644937]
পাকিস্তান	[0.51965935, 0.66609699, 0.07618224, 0.56422173, 0.47343341, 0.38467885, 0.29864259, 0.88069674, 0.42690365, 0.68592679]
রক্ষা	[0.05630811, 0.03494023, 0.67966712, 0.35728259, 0.48713973, 0.2572023, 0.22412101, 1.06173087, 0.12335509, 0.56994076]
অজুহাত	[0.94283084, 0.00690432, 0.16320802, 0.42800279, 0.90944612, 0.64840613, 1.01813438, 0.11089753, 0.25349856, 0.757886]
ঢাকা	[0.22199542, 0.60316483, 0.10002558, 0.1505627, 0.7741713, 0.25723261, 0.17216861, 0.00254664, 0.87566973, 0.45199694]

Table 4: Example of embedding vector of words

5.3 Sentence Embedding

Vector representation of words can be used to generate sentence embedding. Average value of each word can be used to generate initial sentence representation but this variable length representation is not useful for feeding another classifier network [10] [13][. So an encoder model with two decoder model will be trained to generate fixed length sentence representation of each sentence [11] [15]. The encoder will encode each sentence, the first decoder will predict previous sentence of the document and another decoder will predict the next sentence of the document. After completion of the model both decoders will be discarded and the encoder model will be saved to generate fixed length sentence embedding.

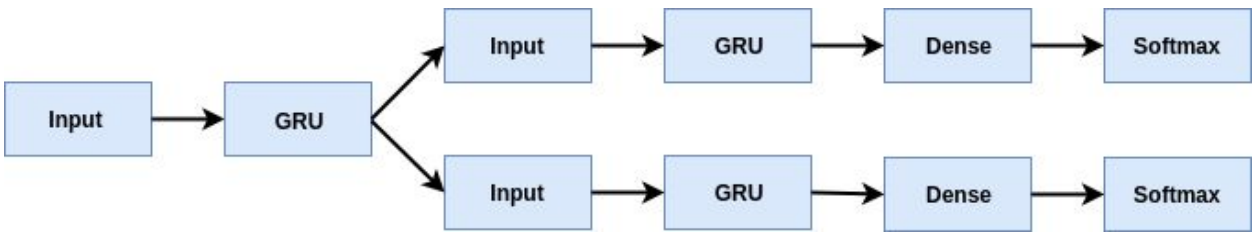


Figure 6: Sentence embedding model architecture

5.4 Feature Extraction

Generation of embedding is useful to represent semantical meaning of a sentence but not enough to determine whether a sentence contains a portion of the main theme of a document. Several other important features must be considered [Kosrow, 2004]. They are ,

- Sentence length
- Modified sentence location
- Degree of centrality
- Keyword frequency
- Similarity to first sentence
- Summation of term frequencies of words

The sentences that occur in the beginning and the conclusion part of the document are most likely important since most documents are hierarchically structured with important information in the beginning and the end of the paragraphs.

Sentence location is calculated as,

$$modified\ sentence\ location\ = 1 - (sentence\ location \div document\ length)$$

Degree of centrality or sentence to sentence cohesion [Gunes, 2000] can be calculated as,

1. Calculate similarity of each sentence with other sentences of the document.
2. Apply a threshold value on each similarity value, set the value to 1 if it is higher than threshold else 0.
3. Add all one of a sentence which represents its centrality value.

Sentence similarity equation ,

$$\text{idf-modified-cosine}(x, y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x}\text{tf}_{w,y}(\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x}(\text{tf}_{x_i,x}\text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y}(\text{tf}_{y_i,y}\text{idf}_{y_i})^2}}$$

Figure 6: Sentence similarity equation

Term frequency equation:

$$tf(t) = number\ of\ terms\ appears\ in\ a\ document\ \div total\ number\ of\ terms\ in\ the\ document$$

$$idf(t)\ = (total\ number\ of\ sentences\ \div number\ of\ sentences\ with\ term\ t\ in\ it)$$

Example:

Consider the article below,

অখণ্ড পাকিস্তান রক্ষার অজুহাতে ঢাকা আজ ধ্বংসপ্রাপ্ত ও ভয়ের নগরী। ঠান্ডা মাথায় পাকিস্তানি সৈন্যরা টানা ২৪ ঘন্টা গোলাবর্ষণের পর নগরীর সাত হাজার মানুষ নিহত, বিস্তৃত এলাকা মাটির সঙ্গে মিশে গেছে । পূর্ব পাকিস্তানের স্বাধীনতার লড়াইকে নির্মমভাবে থামিয়ে দেওয়া হয়েছে। প্রেসিডেন্ট ইয়াহিয়া খান দাবি করেছেন বটে যে পরিস্থিতি এখন শান্ত, তবু রাস্তাঘাটে দেখা যাচ্ছে গ্রামমুখী হাজার হাজার মানুষ। শহরের রাস্তাঘাট ফাঁকা এবং প্রদেশের অন্যান্য স্থানে হত্যাযজ্ঞ অব্যাহত রয়েছে তবে সন্দেহ নেই যে ট্যাংকের মদদপুষ্ট সৈন্যদল শহর ও অন্যান্য লোকালয় নিয়ন্ত্রণ করছে এবং তাদের বিরুদ্ধে প্রতিরোধ খুব কমই। যেটুকু প্রতিরোধ এখানে-সেখানে চলছে, সেটুকু কার্যকর নয়। কারণে-অকারণে গুলি করে মানুষ মারা হচ্ছে, বাড়িঘর নির্বিচারে গুঁড়িয়ে দেওয়া হচ্ছে। সৈন্যদের দেখে মনে হচ্ছে পূর্ব পাকিস্তানের সাত কোটি মানুষের ওপর তাদের নিয়ন্ত্রণ প্রতিদিনই সুপ্রতিষ্ঠিত হচ্ছে। ঠিক কত নিরীহ মানুষ এ পর্যন্ত জীবন দিয়েছে, সে হিসাব করা খুবই কঠিন। চট্টগ্রাম, কুমিল্লা, যশোর ও ঢাকার হিসাব যোগ করলে এ সংখ্যা ১৫ হাজারে দাঁড়াবে। যা মাপা যায়, তা হলো সামরিক অভিযানের ভয়াবহতা। ছাত্রদের হত্যা করা হয়েছে তাদের বিছানায়, কসাই নিহত হয়েছে তার ছোট্ট দোকানটিতে, নারী ও শিশু ঘরের ভেতর জীবন্ত দগ্ধ হয়েছে, হিন্দু ধর্মাবলম্বী পাকিস্তানিদের একসঙ্গে জড়ো করে মারা হয়েছে, বাড়িঘর-বাজার-দোকানপাট জ্বালিয়ে দেওয়া হয়েছে আর পাকিস্তানি পতাকা উড়ছে সব ভবনের শীর্ষে। সরকারি সৈন্যদের দিকে হতাহতের সংখ্যা স্পষ্ট নয় তবে শোনা যাচ্ছে, একজন অফিসার নিহত এবং দুজন সৈন্য আহত হয়েছে।

Sentence similarity table of the document above,

id	1	2	3	4	5	6	7	8	9	10	11
1	1.00	0.45	0.02	0.17	0.03	0.22	0.05	0.028	0.06	0.06	0.00
2	0.45	1.00	0.16	0.27	0.03	0.19	0.03	0.21	0.03	0.15	0.00
3	0.02	0.16	1.00	0.03	0.00	0.01	0.03	0.04	0.00	0.01	0.00
4	0.17	0.27	0.03	1.00	0.01	0.16	0.28	0.17	0.00	0.09	0.01
5	0.03	0.03	0.00	0.01	1.00	0.29	0.05	0.15	0.20	0.04	0.18
6	0.22	0.19	0.01	0.16	0.29	1.00	0.05	0.29	0.04	0.20	0.03
7	0.03	0.03	0.03	0.28	0.05	0.05	1.00	0.06	0.00	0.00	0.01
8	0.28	0.21	0.04	0.17	0.15	0.29	0.06	1.00	0.25	0.20	0.17
9	0.06	0.03	0.00	0.00	0.20	0.04	0.00	0.25	1.00	0.26	0.28
10	0.06	0.15	0.01	0.09	0.04	0.20	0.00	0.20	0.26	1.00	0.12
11	0.00	0.00	0.00	0.01	0.18	0.03	0.01	0.17	0.38	0.12	1.00

Table 5: Cosine similarity matrix of sentences of a document.

Above matrix can be used to calculate the degree of centrality of each sentence.

ID	Degree (0.1)	Degree (0.2)	Degree (0.3)
1	5	4	2
2	7	4	2
3	2	1	1
4	6	3	1
5	5	2	1
6	7	5	1
7	2	2	1
8	9	6	1
9	5	4	2
10	6	4	1
11	5	2	2

Table 6: Calculation of degree of centrality from similarity matrix

Finally if we collect all the extracted features then assuming threshold of degree of centrality is 0.20 and top 15 words with highest frequency as keywords ,

ID	Length	Modified location	Degree of centrality (0.20)	Keyword frequency	Similarity to first sentence	Summation of term frequency
1	10	.91	4	6	1.00	10
2	23	.82	4	12	0.45	32
3	18	.73	1	8	0.02	16
4	13	.64	3	2	0.17	18
5	12	.55	2	3	0.03	12
6	14	.46	5	1	0.22	17
7	10	.37	2	9	0.05	16
8	12	.27	6	7	0.028	14
9	10	.18	4	5	0.06	10
10	11	.09	4	4	0.06	11
11	18	.11	2	2	0.00	24

Table 7: Calculation of all features together

5.5 Model Training and Summary Generation

After generating complete features a binary classifier neural network model will be trained and saved to disk. For generating summary at first the model will be loaded from disk. Each sentence will be converted to vector representation following steps discussed as previous points then the loaded classifier model will be used to predict probabilities of each sentence to be included into summary. After getting probabilities of each sentence there are two choices to generate summary.

- Extract first n sentence with higher score value [4]. This approach extracts the main highlighted portion of the document.

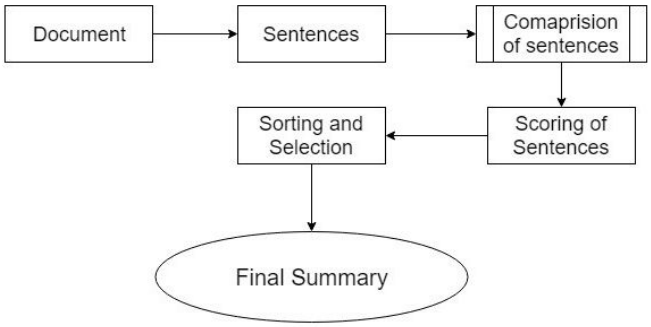


Figure 7: Regular method of generating summary.

- Divide the document or article into pages where length of each page is same and pick up the sentence with most score value from each page [3][5][10]. This approach extracts all topics of the document.

.

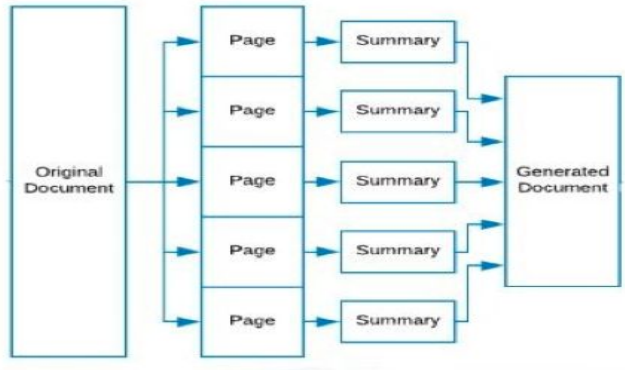


Figure 8: Pagination method of generating summary.

Chapter 6

Class Based Modeling Concept

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

6.1 User Story

User will download the installer and after installation of the installer a graphical user interface will pop up with a text box. The user will paste original document into it and provide ratio of the summary and original document by sliding a slide bar, after clicking the ‘summarize’ button an extractive summary will be generated inside a text field beside the input text box. Summary generation steps are as follows, preprocessing, word embedding by using Word2Vec model, sentence embedding using encoder-decoder sequence model, collections of other features such as length, positional value, degree of centrality, number of cue words, similarity to tittle. Then a final neural network model will be generated to predict probabilities of sentences to be included or not included in the summary.

6.2 General Classifications

To identify the potential class, we have to first select the nouns from the solution space of the story. These were then characterized in seven general classifications. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Following are the specifications of the nouns according to the general classifications:

Serial no	Noun	Problem Domain	General Classification
1	Document	s	5
2	Word	s	2
3	Length	s	2
4	Sentence	s	2
5	Encoder	s	5
6	Decoder	s	5
7	Embedding	s	5
8	classification	p	1
9	classifier	p	1
10	ratio	s	1
11	vector	s	2
12	cue	s	2
13	model	s	2
14	tittle	s	2
15	position	s	2
16	feature	s	2
17	position	s	2
18	Preprocess	s	4
19	Tokenization	s	4
20	Stemming	s	4

Table 8: General Classification

6.3 Selection Criteria

Six selection characteristics should be considered for each potential class for inclusion in final class. They are:
1. Retained information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations

Serial no	Noun	Selection Criteria
1	Preprocess	1,2
2	Word Embedding	1,2
3	Sentence Embedding	1,2
4	Features	1,2
5	Model	1,2

Table 9: Selection Criteria

6.4 Class Cards

Preprocess	
Attributes	Methods
	read_documents() tokenize_words() stem_words() remove_stopwords()

Word_Embedding	
Attributes	Methods

Embedding_length Window_size Iteration_number Learning_rate min_count	fit_model() embed_word()

Sentence_Embedding	
Attributes	Methods
Embedding_length	fit_model() embed_sentence()

Features_Extractor	
Attributes	Methods
Lengths Degrees Positions key_word_length	count_len() count_degree() get_position()

Summarizer	
Attributes	Methods
summary_ratio	get_summary_ratio() generate_summary()

Chapter 7

Architectural Design

As architectural design begins, the software to be developed must be put into context that is, the design should define the external entities (other systems, devices, people) that the software interacts with and the nature of the interaction. This information can generally be acquired from the requirements model and all other information gathered during requirements engineering. Once context is modeled and all external software interfaces have been described, it is easy to identify a set of architectural archetypes.

7.1 Representing System in Context

At the architectural design level, a software architect uses an architectural context diagram (ACD) to model the manner in which software interacts with entities external to its boundaries. systems that interoperate with the target system (the system for which an architectural design is to be developed) are represented as

- Superordinate systems
- Subordinate systems
- Peer-level systems

The following diagram represents the software in context

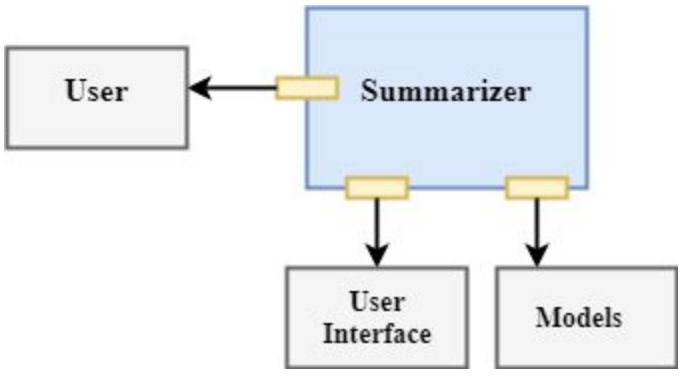


Figure 9: System Architecture in Context

7.2 Refine the architecture into components

As the software architecture is refined into components, the structure of the system begins to emerge. The analysis classes introduced in software requirement modeling represent entities within the application domain that must be addressed within the software architecture. Hence, the application domain is one source for the derivation and refinement of components. Another source is the infrastructure domain. The architecture must accommodate many infrastructure components that enable application components but have no business connection to the application domain. The interfaces depicted in the architecture context diagram imply one or more specialized components that process the data that flows across the interface For the proposed web tool the following components can be introduced:

- Data reading
- Classifier model creation
- Summarizing document

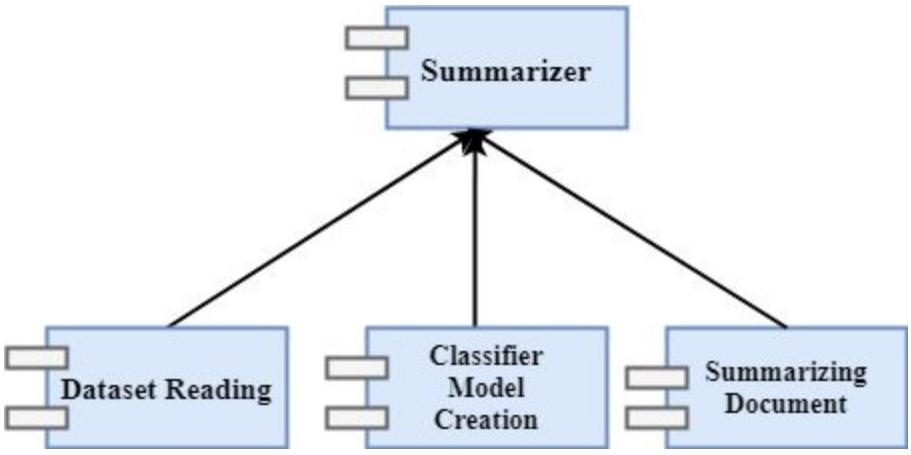


Figure 10: Components of the system

Chapter 8
User Interface Design

User Interface Design is the design of websites, computers, appliances, machines, mobile communication devices, and software applications with a focus on the user's experience and interaction.

Interface Analysis

We divide interface analysis into following parts:

1. User Analysis
2. Task Analysis

User Analysis

In this part we follow two steps:

a) Identify user
b) Know user

Identify user

From the requirements specification we have identified following one user categories.

1. Normal User

Know user

Normal User
Age: Varies
Skills: Varies
Domain expert: Varies
Frequency of use: Frequently
Consequence of a mistake: Medium
General computer experience: Varies

Task Analysis

Normal User: User has following tasks.

1. Summary generation
Goal: Want to generate summary of a single document.
Precondition: User must input a valid dataset
Sub-task: Install the software

8.1 Depiction of Interface States to the End User

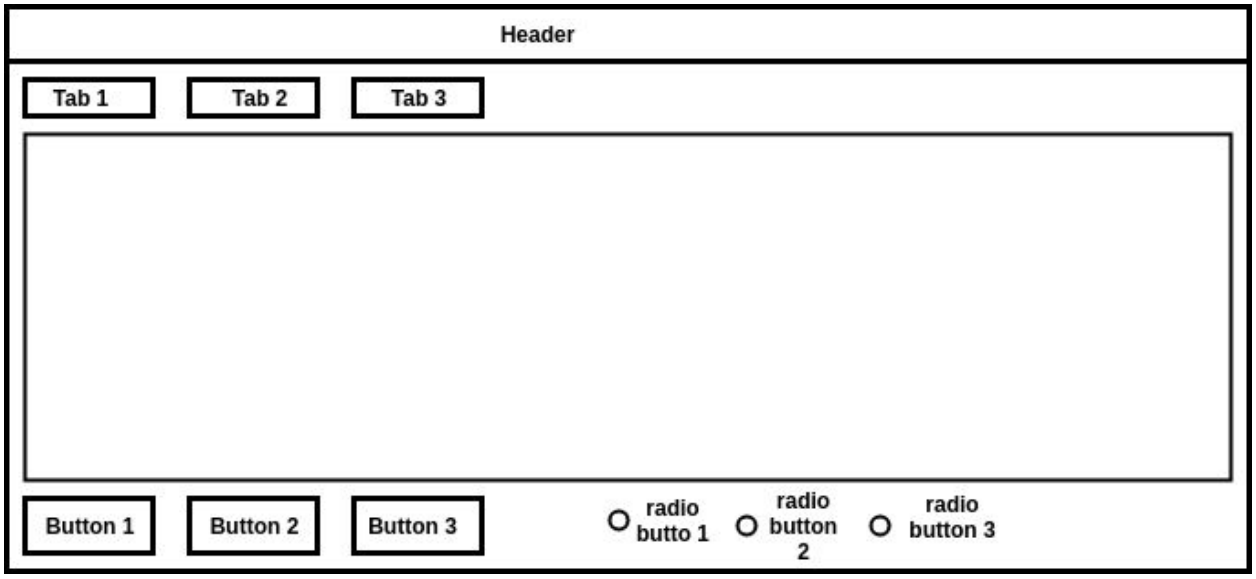


Fig 11: Home page

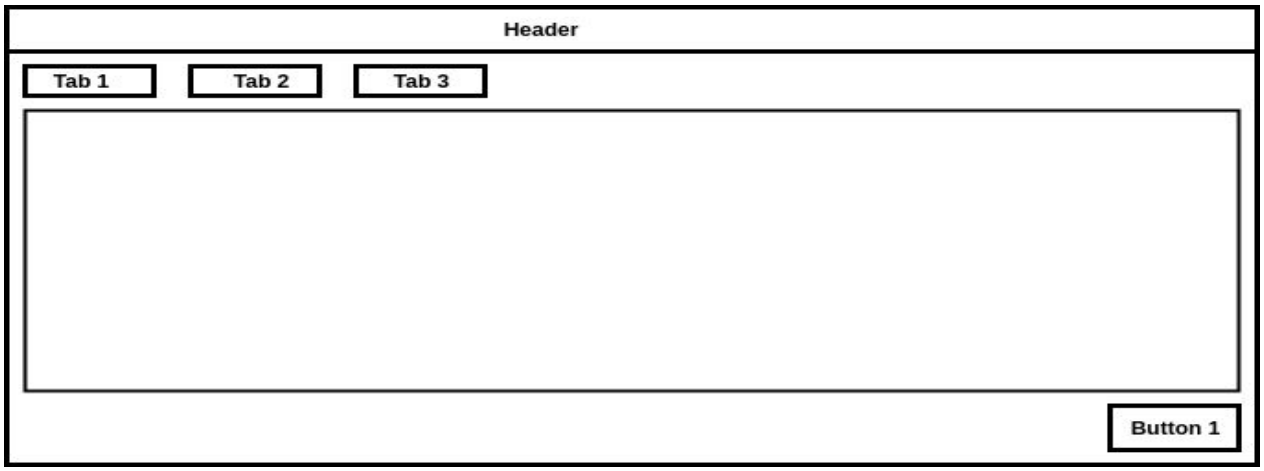


Fig 12: Design of tab 1, tab 2 and tab 3

8.2 Depict Each Interface State as It Looks to End User

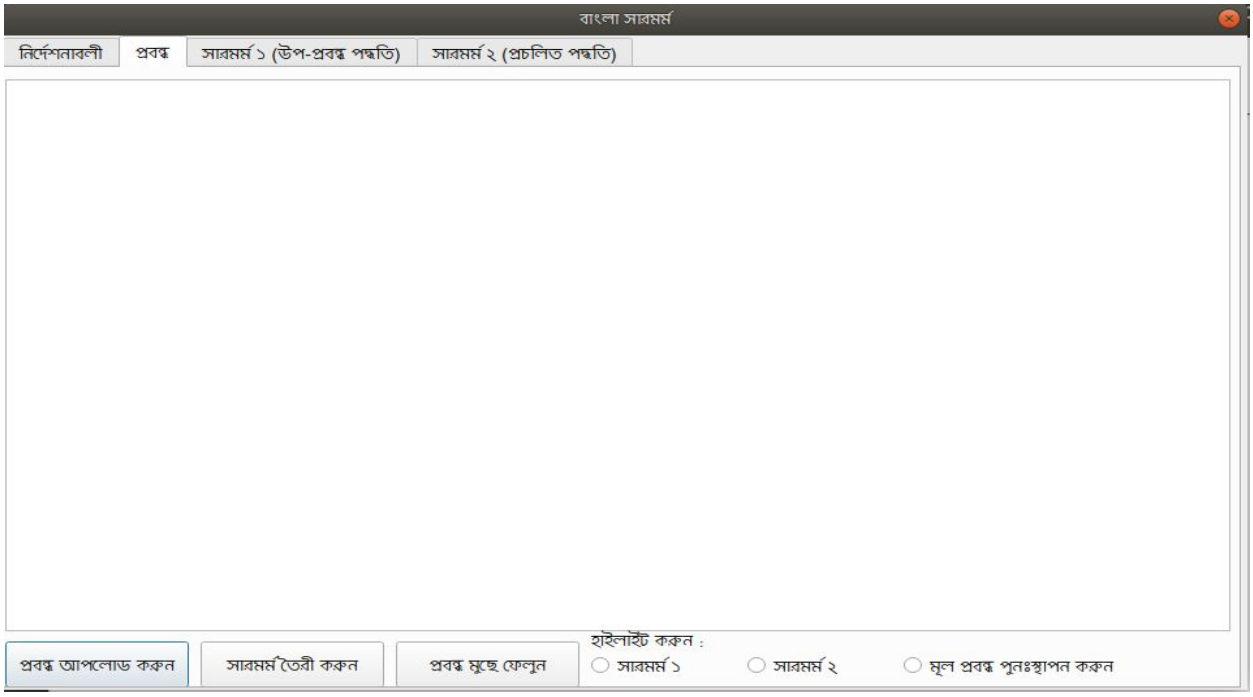


Fig 13 : Home page interface

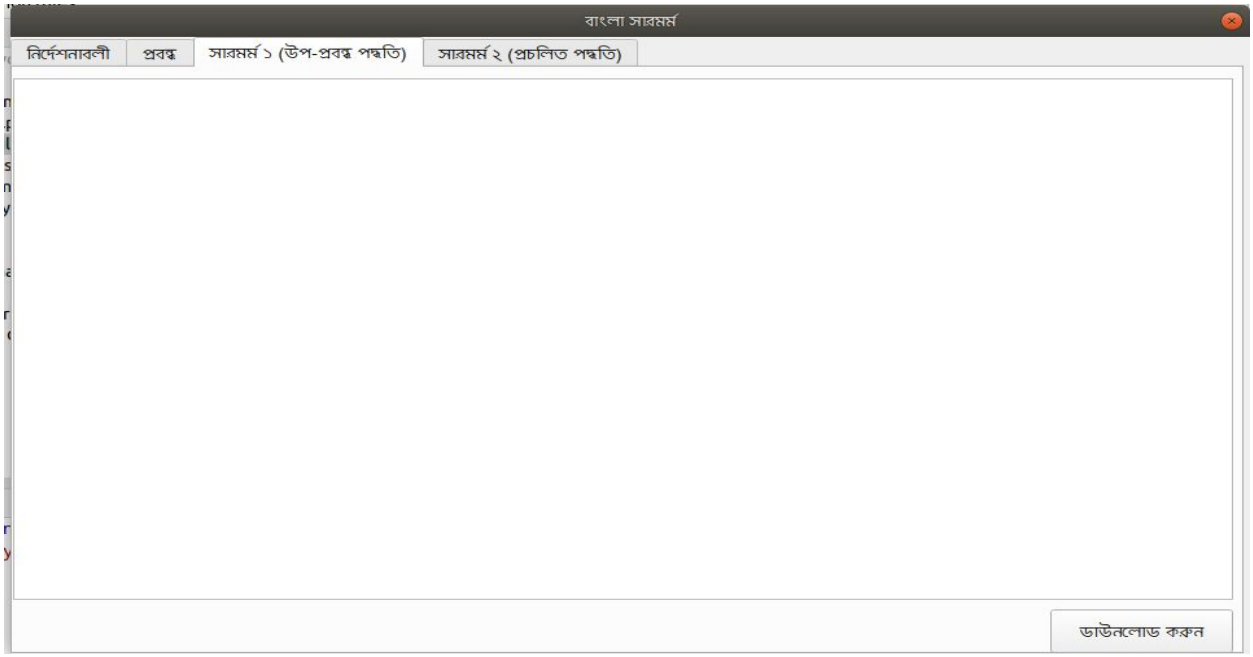


Fig 14: Tab page interface

Chapter 9

Evaluation and Results

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [Lin, 2005] is a metric system to compare machine generated summaries or translation against a reference summaries (Normally human generated summary by manual). ROUGE tends to generate a metric value that determines the accuracy of the generated summary by generating a ratio of overlapping sentences.

For the evaluation of the system's summary generated from the 2 different methods, the ROUGE-2 measure was used. What it does is, it compares the summary generated by the system with the human produced summary. It has two criteria for evaluation:

1. Recall
2. Precision

Recall finds out if the system summary has sentences which match with the reference summary or not. It uses the following formula for computation:

$$Recall = \frac{Number\ of\ overlapping\ Sentences}{Total\ number\ of\ Sentences\ in\ reference\ summary}$$

A perfect score of 1 would mean the machine generated summary matched fully with the reference summary. However the system summary might have useless and unnecessary information in addition to the information presented in the reference summary, and still, recall would give a good score. A better way to see if in fact only the relevant information is present in the system summary or not is by using precision measure.

Precision measure finds out how much of the reference summary is actually present in the system summary by the following formula:

$$Precision = \frac{Number\ of\ overlapping\ Sentences}{Total\ number\ of\ Sentences\ in\ system\ summary}$$

It simply finds out if the system summary is indeed relevant and concise or not.

Lastly, the F1 measure which is a measure of a test's accuracy is calculated using both recall and precision values. A score of 0 means the test yielded the worst results while 1 stands for the best. According to the system, a score of 1 means the system summary matched exactly with the gold summary while 0 means the system summary is totally inaccurate. The F1 measure is calculated by the following formula:

$$F1 = 2 * \frac{precision * recall}{precison + recall}$$

Here, the best F1 score was found at, degree of centrality threshold =0.02, learning rate =0.01 and summary ratio =0.3. 10 fold cross validation technique was to validate the final classifier neural model. Overall dataset set was split into 80-20 training and test set. Training dataset contains 800 documents and testing dataset contains 200 documents. Table 10 shows calculated F1 values for traditional techniques, figure 12 shows F1 values of each cross validation step. Table 11 shows calculated F1 values for techniques, figure 12 shows F1 values of each cross validation step for sub article technique.

Fold number	F1 value
1	61.01
2	64.00
3	63.11
4	67.09
5	68.99
6	65.69
7	69.02
8	65.88
9	67.03
10	69.69

Table 10: Calculation of F1 value for traditional approach

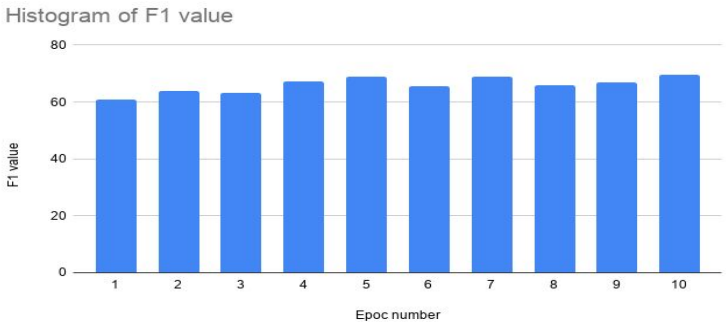


Figure 15: F1 distribution for regular approach

Fold number	F1 value
1	61.01
2	68.39
3	64.81
4	70.87
5	71.32
6	68.98
7	67.46
8	67.38
9	65.98
10	66.92

Table 11: Calculation of F1 value for pagination approach

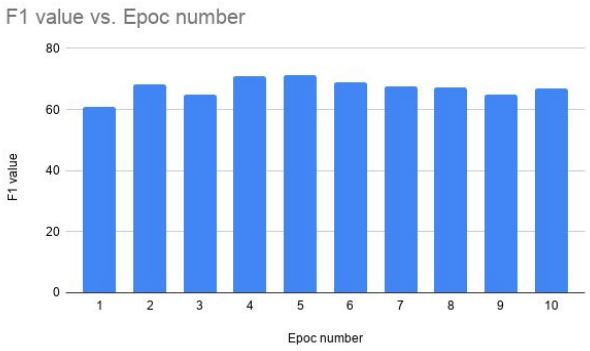


Figure 16: F1 distribution for sub article approach

Test Example 1

Considering the following article,

অখণ্ড পাকিস্তান রক্ষার অজুহাতে ঢাকা আজ ধ্বংসপ্রাপ্ত ও ভয়ের নগরী। ঠান্ডা মাথায় পাকিস্তানি সৈন্যরা টানা ২৪ ঘন্টা গোলাবর্ষণের পর নগরীর সাত হাজার মানুষ নিহত, বিশাল বিস্তৃত এলাকা মাটির সঙ্গে মিশে গেছে এবং পূর্ব পাকিস্তানের স্বাধীনতার লড়াইকে নির্মমভাবে থামিয়ে দেওয়া হয়েছে।প্রেসিডেন্ট ইয়াহিয়া খান দাবি করেছেন বটে যে পরিস্থিতি এখন শান্ত, তবু রাস্তাঘাটে দেখা যাচ্ছে গ্রামমুখী হাজার হাজার মানুষ। শহরের রাস্তাঘাট ফাঁকা এবং প্রদেশের অন্যান্য স্থানে হত্যাযজ্ঞ অব্যাহত রয়েছে ।তবে সন্দেহ নেই যে ট্যাংকের মদদপুষ্ট সৈন্যদল শহর ও অন্যান্য লোকালয় নিয়ন্ত্রণ করছে এবং তাদের বিরুদ্ধে প্রতিরোধ খুব কমই। যেটুকু প্রতিরোধ এখানে-সেখানে চলছে, সেটুকু কার্যকর নয়। কারণে-অকারণে গুলি করে মানুষ মারা হচ্ছে, বাড়িঘর নির্বিচারে গুঁড়িয়ে দেওয়া হচ্ছে। সৈন্যদের দেখে মনে হচ্ছে পূর্ব পাকিস্তানের ৭ কোটি ৩০ লাখ মানুষের ওপর তাদের নিয়ন্ত্রণ প্রতিদিনই সুপ্রতিষ্ঠিত হচ্ছে। ঠিক কত নিরীহ মানুষ এ পর্যন্ত জীবন দিয়েছে, সে হিসাব করা খুবই কঠিন। চট্টগ্রাম, কুমিল্লা, যশোর ও ঢাকার হিসাব যোগ করলে এ সংখ্যা ১৫ হাজারে দাঁড়াবে। যা মাপা যায়, তা হলো সামরিক অভিযানের ভয়াবহতা। ছাত্রদের হত্যা করা হয়েছে তাদের বিছানায়, কসাই নিহত হয়েছে তার ছোট্ট দোকানটিতে, নারী ও শিশু ঘরের ভেতর জীবন্ত দগ্ধ হয়েছে, হিন্দু ধর্মাবলম্বী পাকিস্তানিদের একসঙ্গে জড়ো করে মারা হয়েছে, বাড়িঘর-বাজার-দোকানপাট জ্বালিয়ে দেওয়া হয়েছে আর পাকিস্তানি পতাকা উড়ছে সব ভবনের শীর্ষে। সরকারি সৈন্যদের দিকে হতাহতের সংখ্যা স্পষ্ট নয় তবে শোনা যাচ্ছে, একজন অফিসার নিহত এবং দুজন সৈন্য আহত হয়েছে।

Human generated summary:

অখণ্ড পাকিস্তান রক্ষার অজুহাতে ঢাকা আজ ধ্বংসপ্রাপ্ত ও ভয়ের নগরী ঠান্ডা মাথায় পাকিস্তানি সৈন্যরা টানা ২৪ ঘন্টা গোলাবর্ষণের পর নগরীর সাত হাজার মানুষ নিহত, বিশাল বিস্তৃত এলাকা মাটির সঙ্গে মিশে গেছে এবং পূর্ব পাকিস্তানের স্বাধীনতার লড়াইকে নির্মমভাবে থামিয়ে দেওয়া হয়েছে সৈন্যদের দেখে মনে হচ্ছে পূর্ব পাকিস্তানের ৭ কোটি ৩০ লাখ মানুষের ওপর তাদের নিয়ন্ত্রণ প্রতিদিনই সুপ্রতিষ্ঠিত হচ্ছে বাড়িঘর–বাজার–দোকানপাট জ্বালিয়ে দেওয়া হয়েছে আর পাকিস্তানি পতাকা উড়ছে সব ভবনের শীর্ষে।

After feeding the document probabilities of sentences will be,

ID	Probability
1	9.0738773e-01
2	9.0738773e-01
3	9.0399361e-01
4	9.6092522e-03
5	5.2191615e-02
6	5.7975799e-02
7	2.3657084e-04
8	1.2747943e-03
9	7.4560010e-01
10	1.2405030e-02
11	2.8333541e-03
12	2.4690903e-03

Table 12: Probability of sentences for document 1

System generated summary 1 (Traditional method):

অখণ্ড পাকিস্তান রক্ষার অজুহাতে ঢাকা আজ ধ্বংসপ্রাপ্ত ও ভয়ের নগরী। সৈন্যদের দেখে মনে হচ্ছে পূর্ব পাকিস্তানের ৭ কোটি ৩০ লাখ মানুষের ওপর তাদের নিয়ন্ত্রণ প্রতিদিনই সুপ্রতিষ্ঠিত হচ্ছে। চট্টগ্রাম, কুমিল্লা, যশোর ও ঢাকার হিসাব যোগ করলে এ সংখ্যা ১৫ হাজারে দাঁড়াবে। সরকারি সৈন্যদের দিকে হতাহতের সংখ্যা স্পষ্ট নয় তবে শোনা যাচ্ছে, একজন অফিসার নিহত এবং দুজন সৈন্য আহত হয়েছে।

System generated summary 2 (Pagination or sub documentation method):

অখণ্ড পাকিস্তান রক্ষার অজুহাতে ঢাকা আজ ধ্বংসপ্রাপ্ত ও ভয়ের নগরী। সৈন্যদের দেখে মনে হচ্ছে পূর্ব পাকিস্তানের ৭ কোটি ৩০ লাখ মানুষের ওপর তাদের নিয়ন্ত্রণ প্রতিদিনই সুপ্রতিষ্ঠিত হচ্ছে। ঠিক কত নিরীহ মানুষ এ পর্যন্ত জীবন দিয়েছে, সে হিসাব করা খুবই কঠিন। সরকারি সৈন্যদের দিকে হতাহতের সংখ্যা স্পষ্ট নয় তবে শোনা যাচ্ছে, একজন অফিসার নিহত এবং দুজন সৈন্য আহত হয়েছে।

Test Example 2

Considering article 2

মজুত চাল নিয়ে বিপাকে সরকার। এখন বাজারে মোটা চালের কেজি ৩০ থেকে ৩২ টাকা। আর খাদ্য অধিদপ্তর খোলাবাজারে বিক্রি করছে ২০ টাকায়। কিন্তু এই চালের মান পড়ে যাওয়ায় সহজে বিক্রি হচ্ছে না। ৩২ টাকা কেজি দরে কেনা এই চাল এখন ১৫ টাকায় বিক্রির পরিকল্পনা করছে খাদ্য মন্ত্রণালয়। এদিকে আমনের ৮৯ হাজার টন চাল সংগ্রহ বাকি আছে। আগামী মে থেকে ১০-১২ লাখ টন বোরো সংগ্রহ শুরু হবে। এ জন্য গুদাম খালি করতে হবে। কিন্তু সরকারের খাদ্য বিক্রির অন্যতম উপায় খোলাবাজারে চাল (ওএমএস) বিক্রি প্রায় বন্ধ হয়ে আছে। সাতটি সামাজিক নিরাপত্তা কর্মসূচিতে বরাদ্দ করা খাদ্যের বণ্টন ও বিক্রিও চলছে শ্লথগতিতে। গত ১৩ নভেম্বর খাদ্য অধিদপ্তর থেকে মন্ত্রণালয়ে পাঠানো প্রতিবেদন অনুযায়ী, গুদামে থাকা চালের বড় অংশের মান ক্রমশ কমছে। ওই প্রতিবেদনে দ্রুত চাল খালাসের নির্দেশনা চাওয়া হয়েছে। এখনো সরকারি গুদামে মান পড়ে যাওয়া এক লাখ পাঁচ হাজার টন চাল রয়ে গেছে। গত বুধবার ত্রাণ মন্ত্রণালয়বিষয়ক সংসদীয় স্থায়ী কমিটির সভায় সাংসদেরা টিআর ও কাবিখায় চাল-গম চান না বলে জানিয়ে দিয়েছেন।তাঁরা সরকারের কাছে সামাজিক নিরাপত্তা কর্মসূচির জন্য নগদ অর্থ চেয়েছেন।

Human generated summary,

মজুত চাল নিয়ে বিপাকে সরকার। ৩২ টাকা কেজি দরে কেনা এই চাল এখন ১৫ টাকায় বিক্রির পরিকল্পনা করছে খাদ্য মন্ত্রণালয়। আগামী মে থেকে ১০-১২ লাখ টন বোরো সংগ্রহ শুরু হবে। এ জন্য গুদাম খালি করতে হবে। এখনো সরকারি গুদামে মান পড়ে যাওয়া এক লাখ পাঁচ হাজার টন চাল রয়ে গেছে। গত বুধবার ত্রাণ মন্ত্রণালয়বিষয়ক সংসদীয় স্থায়ী কমিটির সভায় সাংসদেরা টিআর ও কাবিখায় চাল-গম চান না বলে জানিয়ে দিয়েছেন।

After feeding the document probabilities of sentences will be,

ID	Probability
1	9.07E-01
2	1.33E-03
3	1.47E-03

4	9.07E-01
5	7.38E-01
6	5.87E-02
7	9.07E-01
8	4.34E-07
9	9.07E-01
10	9.07E-01
11	7.30E-01
12	7.38E-01
13	2.33E-01

Table 13: Probability of sentences for document 2

System generated summary 1:

মজুত চাল নিয়ে বিপাকে সরকার। ৩২ টাকা কেজি দরে কেনা এই চাল এখন ১৫ টাকায় বিক্রির পরিকল্পনা করছে খাদ্য মন্ত্রণালয়। এ জন্য গুদাম খালি করতে হবে। সাতটি সামাজিক নিরাপত্তা কর্মসূচিতে বরাদ্দ করা খাদ্যের বণ্টন ও বিক্রিও চলছে শ্লথগতিতে। এখনো সরকারি গুদামে মান পড়ে যাওয়া এক লাখ পাঁচ হাজার টন চাল রয়ে গেছে।

System generated summary 2:

মজুত চাল নিয়ে বিপাকে সরকার ৩২ টাকা কেজি দরে কেনা এই চাল এখন ১৫ টাকায় বিক্রির পরিকল্পনা করছে খাদ্য মন্ত্রণালয়। এ জন্য গুদাম খালি করতে হবে। সাতটি সামাজিক নিরাপত্তা কর্মসূচিতে বরাদ্দ করা খাদ্যের বণ্টন ও বিক্রিও চলছে শ্লথগতিতে। গত ১৩ নভেম্বর খাদ্য অধিদপ্তর থেকে মন্ত্রণালয়ে পাঠানো প্রতিবেদন অনুযায়ী, গুদামে থাকা চালের বড় অংশের মান ক্রমশ কমছে।

Chapter 10

Implementation Overview

This chapter includes a short description of source code files, implementation technique and overview of modules or packages. Python 3 programming language has been used for development of core algorithm. PyQt5 has been used for front-end development. PyCharm IDE (Community version) has been used as a development environment. Git has been used for version controlling. Complete code can be found at https://github.com/sefatemahadi/BTS.

10.1 Code Review

This section has been structured based on the files and folders of the main program. All file names, model names, function and class names are self explanatory.

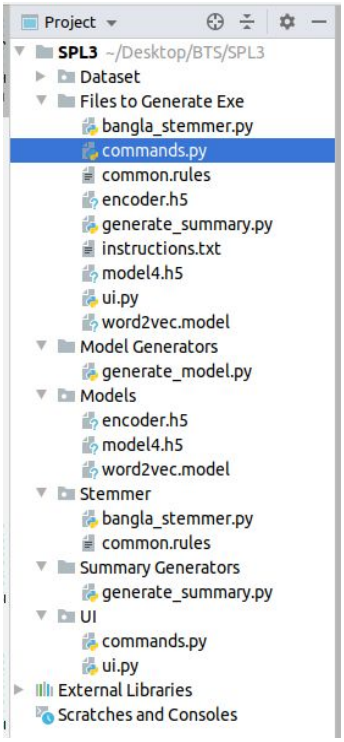


Figure 17: Project hierarchy

Dataset: This directory contains dataset to train neural models. Here the dataset is a collection of several bangla documents along with their corresponding summaries. All documents and summaries are saved into two separate subdirectories named as 'Test' and 'Train'.

Files to generate exe: This directory contains all python scripts, models and textual files to generate a final executable file of the final outcome of the project.

Model Generators: This directory contains files which are necessary to generate word embedding models, sentence embedding models and classifier models.

generate_models.py: This file contains code to generate necessary neural models described in the methodology section. Classes and methods that it contains are create_word_embedding_model, create_sentence_embedding_model, create_model, calculate_common_words, calculate_sentence_frequency, collect_keywords, calculate_degree, calculate_similarity.

Models: This directory contains generated models. These three models are word embedding models, sentence embedding models and classifier models.

- **word2vec.model**: This word embedding models converts each word to a fixed dimension of numeric vectors.
- **encoder.h5**: This sentence embedding model converts each sentence to a fixed dimension numeric vectors.
- **model4.h5**: The classifier model that predicts probability of sentences to be included into automated summary.

Stemmer: This directory contains bangla_stemmer.py script to stemming a bangla word. The stemmer is rule based stemmer and rules are stored in common.rules file.

generate_summary.py: This python file contains codes to collect all features of a document and to generate final summary. Classes and functions of this script are Textline, calculate_sentence_frequency, collect_keywords, calculate_degree, calculate_similarity, load_model, predict.

UI: This directory contains graphical source code of graphical user interface and signals for buttons, radio buttons.

ui.py: It simply contains layout formats, views of user interface.Classes and functions of this file are UI_Dialog, retranslateUI.

Commands.py: Contains necessary signals and commands for buttons of ui.py script. Functions of this script are upload, delete, summarize, mark_lines, download, radio_action, restore_main_article.

10.2 Libraries

This section describes an overview of libraries that have been used to build up the project.

Numpy

NumPy is a library for the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical 22 functions to operate on these arrays.

NLTK

NLTK (Natural Language ToolKit) is the most popular Python framework for working with human language. NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Gensim

Gensim is a Python library designed to automatically extract semantic topics from documents, as efficiently as possible.Gensim is designed to process raw, unstructured digital texts or plain text. Gensim word2vec library provides external support to preprocess and build up word embedding model both in continuous bag of words and skip-gram model.

Keras

Keras is a high-level neural networks API, capable of running on top of Tensorflow, Theano. It enables fast experimentation through a high level, user-friendly, modular and extensible API. Keras can also be run on both CPU and GPU. Keras sequential layer is a sub module of keras layer which enables its user to build up traditional neural models just adding on layer on top others. Keras hides extensive mathematical calculations required to train a model from user.

py-bangla-stemmer

py-bangla-stemmer, a rule based stemmer is a python library to stem bangla a bangla word to its root form.

Scikit-learn

Scikit-learn is a machine learning library for python programming language which offers various important features for machine learning such as classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and is designed to interoperate with the python numerical and scientific libraries like numpy and scipy. Although there are lots of use of scikit-learn and it covers all aspects of machine learning, here in this project it is used for feature scaling.

PyQT5

PyQt5 is cross-platform GUI toolkit, a set of python bindings for Qt v5. One can develop an interactive desktop application with so much ease because of the tools and simplicity provided by this library. A GUI application consists of Front-end and Back-end. PyQt5 has provided a tool called ‘QtDesigner’ to design the front-end by drag and drop method so that development can become faster and one can give more time on back-end stuff.

Chapter 11

User Manual

This chapter contains a user guide, also commonly called a technical communication document or manual, is intended to give assistance to non-technical people using this system.

Intended audience

This document is intended to be used by general people who have minimum knowledge about using a computer. This also helps researchers who are willing to work with Bengali NLP.

System requirements

Computer with the following installed on it:

- Ubuntu 12.03 or later
- Minimum 256 MB of RAM

Run the exe file

To launch the home page go to this link, download the exe file and run it. After extracting, go inside the directory.

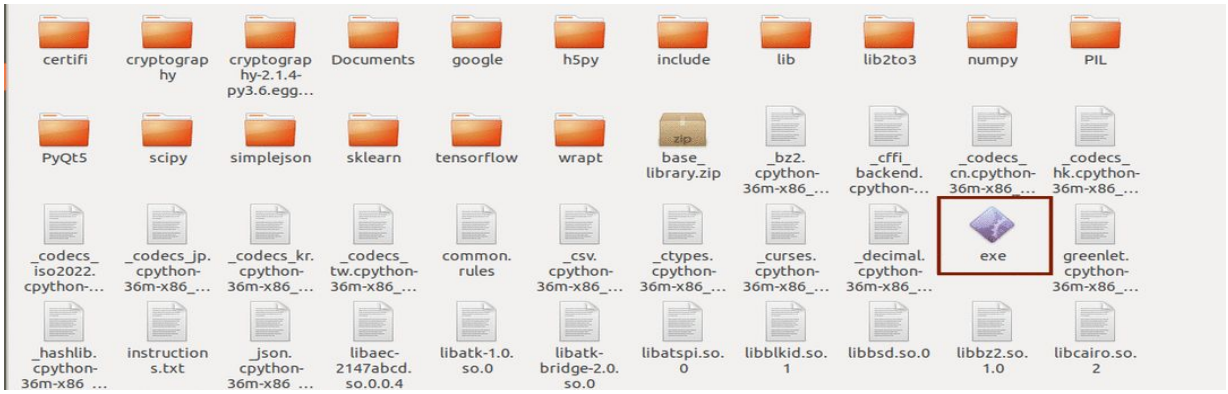


Figure 18: Extraction of the downloaded file

Now open a terminal and run the exe file.

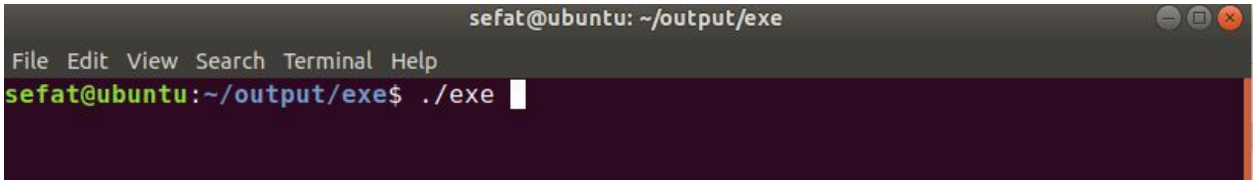


Figure 19: Running the exe

The home page above will look like below.

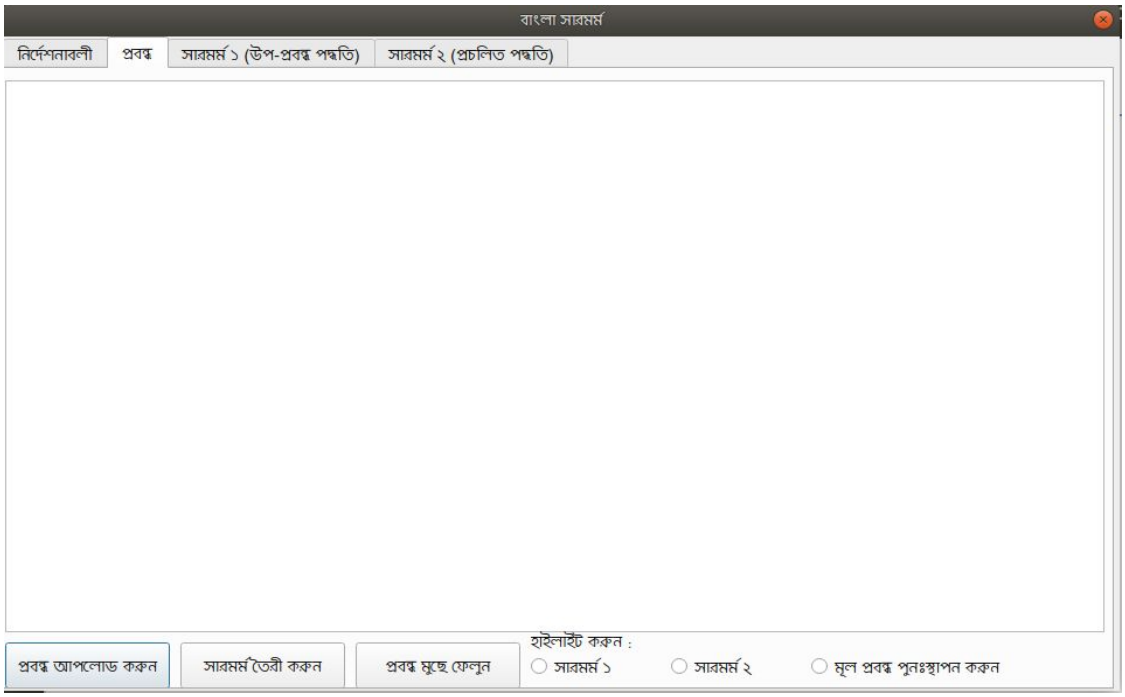


Figure 20: Home page

Read the instructions

To read the instructions click on **"নির্দেশনাবলী"** tab.

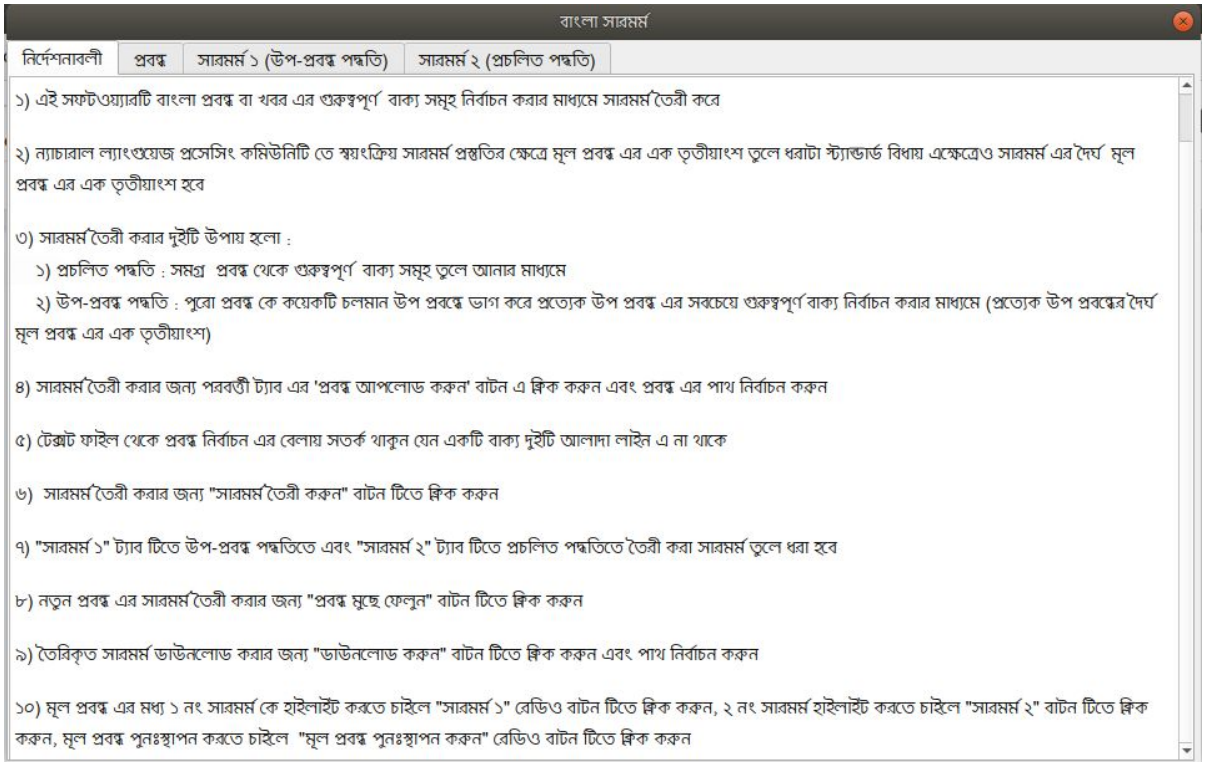


Figure 21: Instruction tab

Upload Document

Now to upload a document click on **"প্রবন্ধ আপলোড করুন"** button.

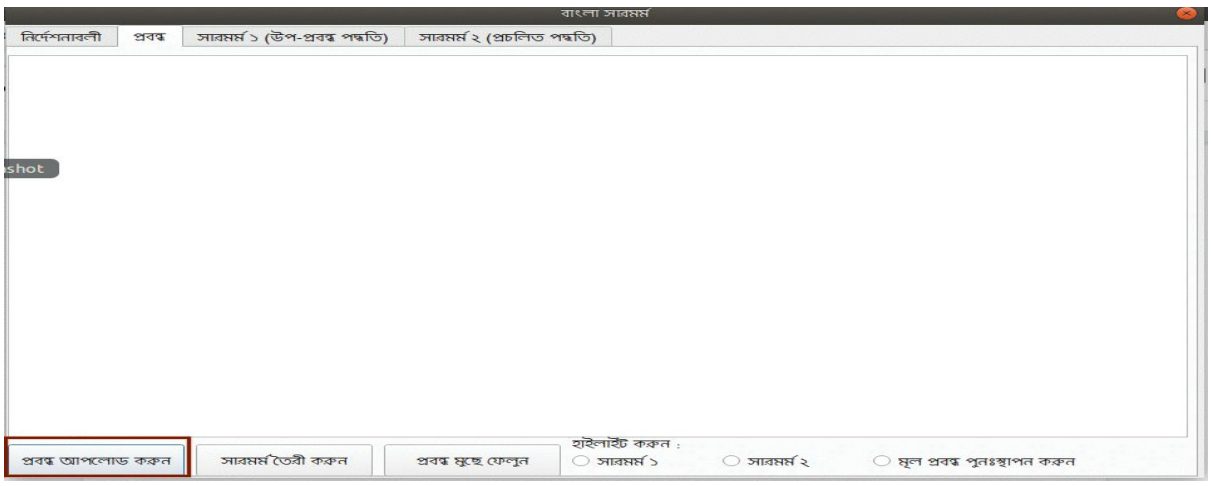


Figure 22: File upload

A pop up with file uploader will appear

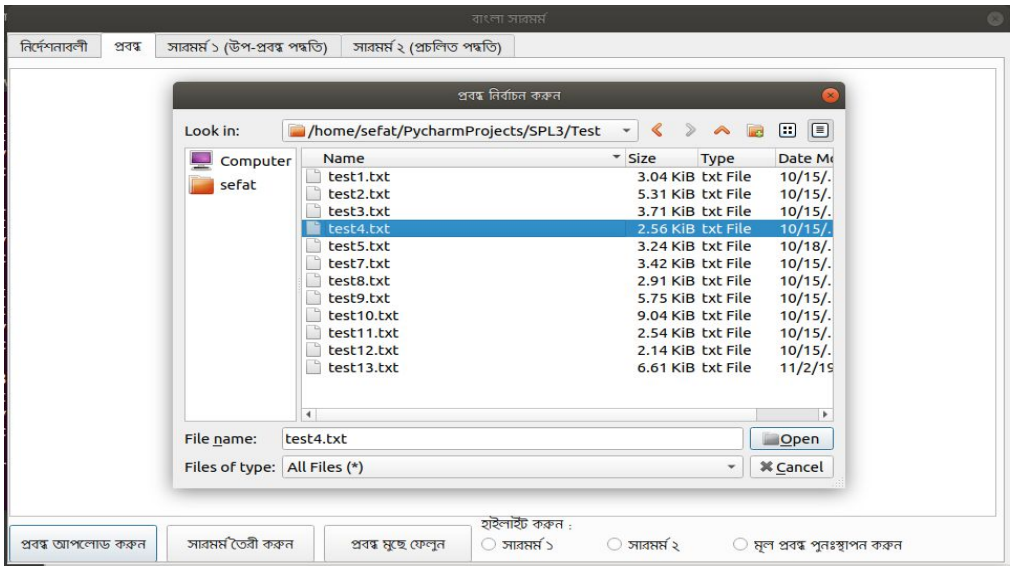


Figure 23: Document selection

After uploading the system will look like this

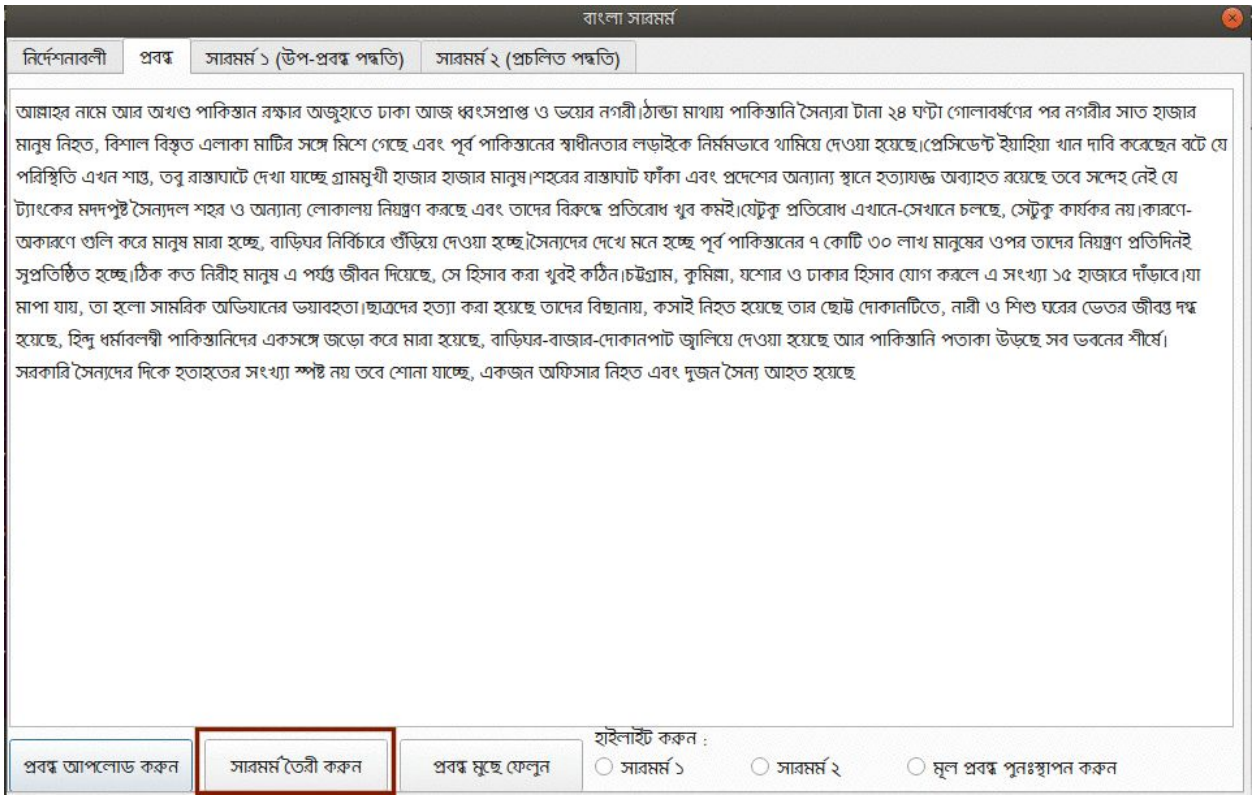


Figure 24: Summary generation

Generate Summary

After uploading the document into system click on **"সারমর্ম তৈরী করুন"** button to generate summary. It might take a few seconds. After generating summary a pop up window will notify the user.

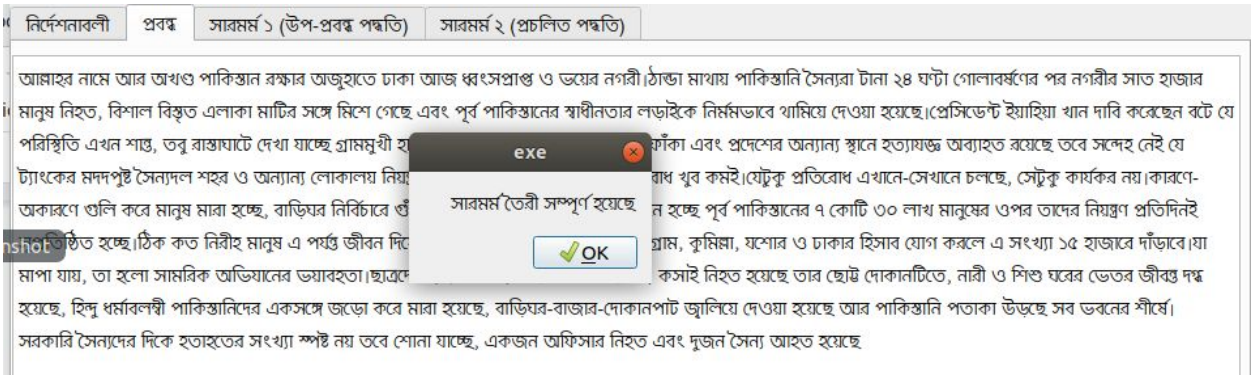


Figure 25: Notification after completion of summary

To view summary with pagination technique click the tab titled as '**সারমর্ম ১ (উপ-প্রবন্ধ পদ্ধতি)**'.

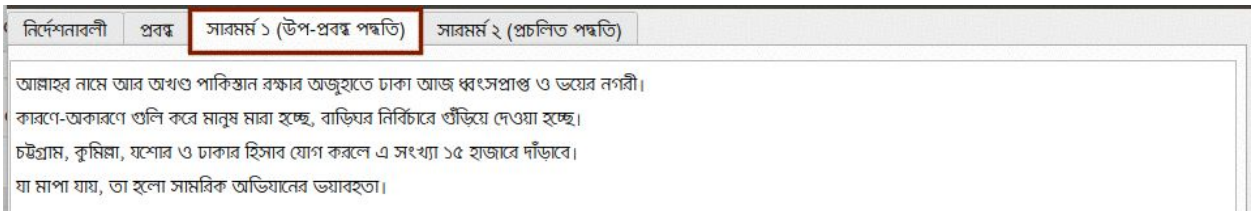


Figure 26: Summary generation in sub article or pagination approach

To view summary without pagination technique click the tab titled as **'সারমর্ম ২ (প্রচলিত পদ্ধতি)'**.

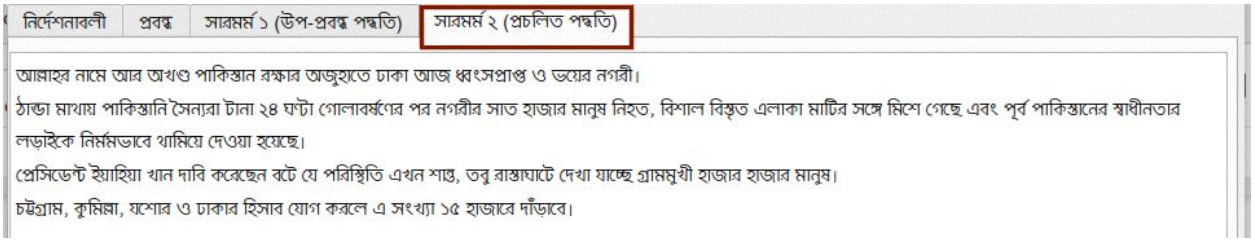


Figure 27: Summary generation in regular approach

Download Summary

To download generated summaries in a text file. Go to the tab and click on 'ডাউনলোড করুন' button.

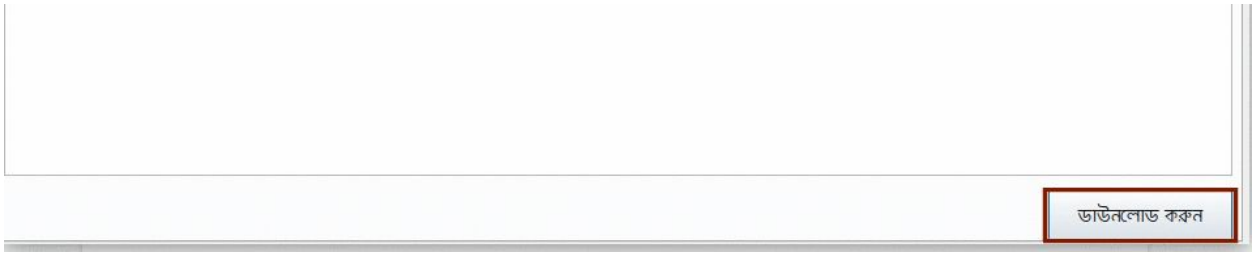


Figure 28: Summary download button.

View Generated Summary in Main Article

This system provides functionality to view generated article inside the main article so to make evaluation of generated summaries easier. To view the summary with pagination technique go home tab and click on radio button named as 'সারমর্ম ১'.

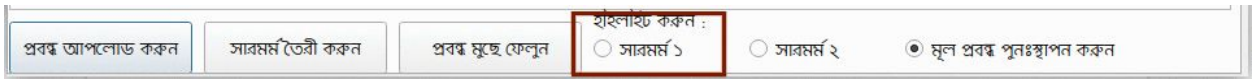


Figure 29: Highlighting summary 1 in main article

Lines which are included into summary one will be highlighted.

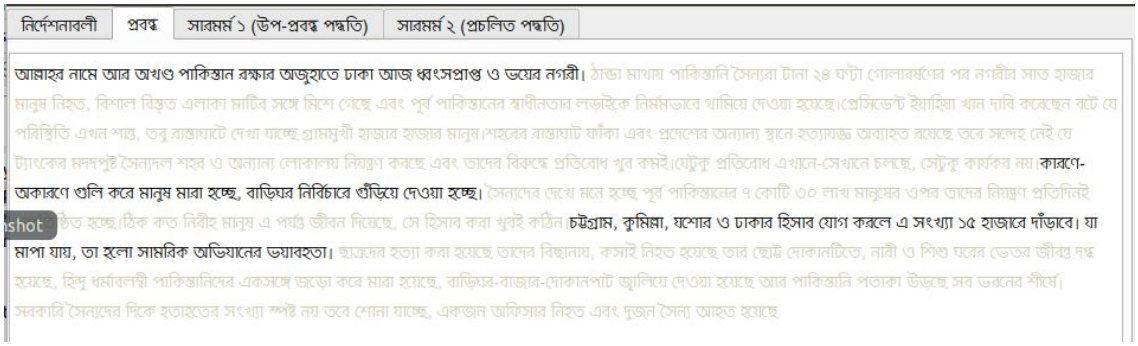
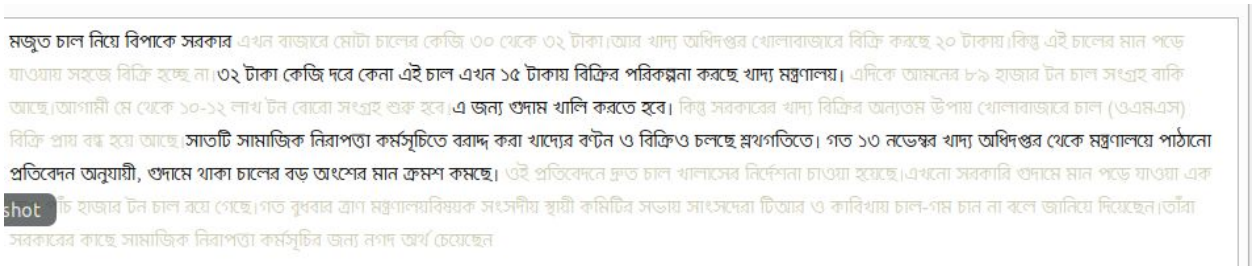


Figure 30: View of summary 1 in main article

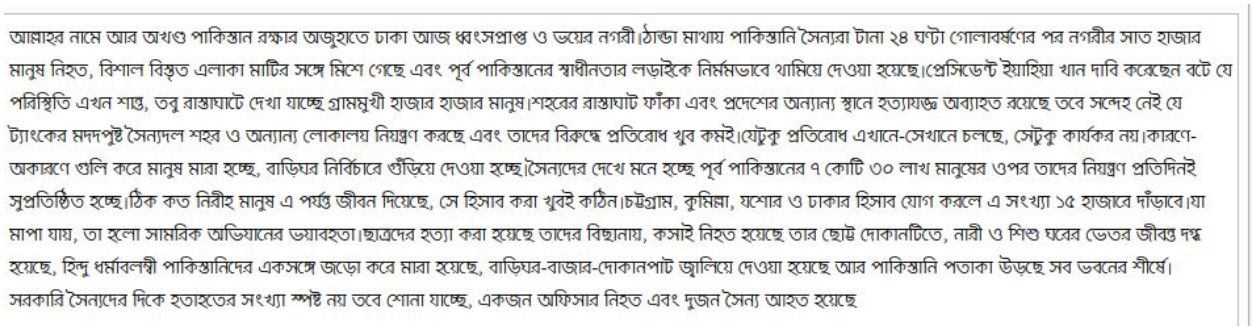
By clicking on radio button 2 named as 'সারমর্ম ২' summary of the third tab will highlighted into the main article.



মজুত চাল নিয়ে বিপাকে সরকার এখন বাজারে মোটা চালের কেজি ৩০ থেকে ৩২ টাকা।আর খাদ্য অধিদপ্তর খোলাবাজারে বিক্রি করছে ২০ টাকায়।কিন্তু এই চালের মান পড়ে যাওয়ায় সহজে বিক্রি হচ্ছে না।৩২ টাকা কেজি দরে কেনা এই চাল এখন ১৫ টাকায় বিক্রির পরিকল্পনা করছে খাদ্য মন্ত্রণালয়। এদিকে আমনের ৮৯ হাজার টন চাল সংগ্রহ বাকি আছে।আগামী মে থেকে ১০-১২ লাখ টন বোরো সংগ্রহ শুরু হবে।এ জন্য গুদাম খালি করতে হবে। কিন্তু সরকারের খাদ্য বিক্রির অন্যতম উপায় খোলাবাজারে চাল (ওএমএস) বিক্রি প্রায় বন্ধ হয়ে আছে।সাতটি সামাজিক নিরাপত্তা কর্মসূচিতে বরাদ্দ করা খাদ্যের বণ্টন ও বিক্রিও চলছে শ্লথগতিতে। গত ১৩ নভেম্বর খাদ্য অধিদপ্তর থেকে মন্ত্রণালয়ে পাঠানো প্রতিবেদন অনুযায়ী, গুদামে থাকা চালের বড় অংশের মান ক্রমশ কমছে। ওই প্রতিবেদনে দ্রুত চাল খালাসের নির্দেশনা চাওয়া হয়েছে।এখনো সরকারি গুদামে মান পড়ে যাওয়া এক shot চি হাজার টন চাল রয়ে গেছে।গত বুধবার ত্রাণ মন্ত্রণালয়বিষয়ক সংসদীয় স্থায়ী কমিটির সভায় সাংসদেরা টিআর ও কাবিখায় চাল-গম চান না বলে জানিয়ে দিয়েছেন।তাঁরা সরকারের কাছে সামাজিক নিরাপত্তা কর্মসূচির জন্য নগদ অর্থ চেয়েছেন

Figure 31: View of summary 2 in main article

To get back the main article click '"**মূল প্রবন্ধ পুনঃস্থাপন করুন**"'



আল্লাহর নামে আর অখণ্ড পাকিস্তান রক্ষার অজুহাতে ঢাকা আজ ধ্বংসপ্রাপ্ত ও ভয়ের নগরী।ঠান্ডা মাথায় পাকিস্তানি সৈন্যরা টানা ২৪ ঘণ্টা গোলাবর্ষণের পর নগরীর সাত হাজার মানুষ নিহত, বিশাল বিস্তৃত এলাকা মাটির সঙ্গে মিশে গেছে এবং পূর্ব পাকিস্তানের স্বাধীনতার লড়াইকে নির্মমভাবে থামিয়ে দেওয়া হয়েছে।প্রেসিডেন্ট ইয়াহিয়া খান দাবি করেছেন বটে যে পরিস্থিতি এখন শান্ত, তবু রাস্তাঘাটে দেখা যাচ্ছে গ্রামমুখী হাজার হাজার মানুষ।শহরের রাস্তাঘাট ফাঁকা এবং প্রদেশের অন্যান্য স্থানে হত্যাযজ্ঞ অব্যাহত রয়েছে তবে সন্দেহ নেই যে ট্যাংকের মদদপুষ্ট সৈন্যদল শহর ও অন্যান্য লোকালয় নিয়ন্ত্রণ করছে এবং তাদের বিরুদ্ধে প্রতিরোধ খুব কমই।যেটুকু প্রতিরোধ এখানে-সেখানে চলছে, সেটুকু কার্যকর নয়।কারণে-অকারণে গুলি করে মানুষ মারা হচ্ছে, বাড়িঘর নির্বিচারে গুঁড়িয়ে দেওয়া হচ্ছে।সৈন্যদের দেখে মনে হচ্ছে পূর্ব পাকিস্তানের ৭ কোটি ৩০ লাখ মানুষের ওপর তাদের নিয়ন্ত্রণ প্রতিদিনই সুপ্রতিষ্ঠিত হচ্ছে।ঠিক কত নিরীহ মানুষ এ পর্যন্ত জীবন দিয়েছে, সে হিসাব করা খুবই কঠিন।চট্টগ্রাম, কুমিল্লা, যশোর ও ঢাকার হিসাব যোগ করলে এ সংখ্যা ১৫ হাজারে দাঁড়াবে।যা মাপা যায়, তা হলো সামরিক অভিযানের ভয়াবহতা।ছাত্রদের হত্যা করা হয়েছে তাদের বিছানায়, কসাই নিহত হয়েছে তার ছোট্ট দোকানটিতে, নারী ও শিশু ঘরের ভেতর জীবন্ত দগ্ধ হয়েছে, হিন্দু ধর্মাবলম্বী পাকিস্তানিদের একসঙ্গে জড়ো করে মারা হয়েছে, বাড়িঘর-বাজার-দোকানপাট জ্বালিয়ে দেওয়া হয়েছে আর পাকিস্তানি পতাকা উড়ছে সব ভবনের শীর্ষে। সরকারি সৈন্যদের দিকে হতাহতের সংখ্যা স্পষ্ট নয় তবে শোনা যাচ্ছে, একজন অফিসার নিহত এবং দুজন সৈন্য আহত হয়েছে

Figure 32: Restoration of the main article

Chapter 12: Test Plans

This chapter presents test plan of the project.

Test-Item to be tested:

I tested the Summarization tool. Software requirement analysis and specification document of the system was used for this purpose.

Approach:

I used end to end automated (integration) testing technique to test the system.

Item Pass/Fail Criteria:

If actual output of a test case does not match with the expected output of the test case, the test case is considered as failed. 100% of all test cases should pass. No failed case should be crucial to the end-user's ability to use the application.

Test Deliverables:

I will deliver test plan document, test case and test report.

The following scenarios are considered as risks for the project:
- Delay in requirements engineering.
- Delay in developing.
- Modification in development technology.

Test id	Test scenario	Test steps	Test data	Executed results	Actual result	Pass/Failed
T01	Check the functionality of the "প্রবন্ধ আপলোড করুন" document	Click on the button and upload a text file with .txt extension	A text file with .txt extension	File successfully and contents are displayed within text	As expected	Passed

				field		
T02	Check the functionality of “সারমর্ম তৈরী করুন” button	Click on the button	Uploaded article	Summaries should be appeared on both other tabs	As expected	Passed
T03	Check the functionality of “প্রবন্ধ মুছে ফেলুন” button	Click on the button	Uploaded article	The main article along with both summaries should be disappeared	As expected	Passed
T04	Check the functionality of ‘সারমর্ম ১’ radio button	Click on the radio button	Uploaded article	Summary one should be highlighted in main article	As expected	Passed
T05	Check the functionality of ‘সারমর্ম ২’ radio button	Click on the radio button	Uploaded article	Summary two should be highlighted in main article	As expected	Passed
T06	Check the functionality of ”ডাউনলোড করুন” button	Click on the button	Generated summaries	Selected summary should be downloaded	As expected	Passed
T07	Check the functionality of "মূল প্রবন্ধ পুনঃস্থাপন করুন” radio	Click on the radio button	Highlighted summaries and main article	Main article should be restored	As expected	Passed

Table 14: Test plans

Chapter 13: Conclusion and Future Work

This documentation includes a complete explanation of deep learning approach for extractive bangla article summarization. In short the approach can be summarised as preprocess, construction of word embedding, sentence embedding, feature extraction ,and classification. Overall F1 score is found 68.91. This research work can be used in several cases, such examples are online web portals can use for representing summary of a long news, scientific journals can also use for representing summary of an article etc. The research work for Bangla is difficult as there is hardly any automated tool to facilitate research work, no database for ontological knowledge of words and limited scope of knowledge sharing. Despite these difficulties a new method for bangla text summarization has been introduced. Due to difficulties, complexities of bangla language and lack of necessary resources performance of this proposed method may not be the same as its English counterpart. In future it will be tried to extend this approach to generate abstract summary.

References

1. Aditya, J., Divij, B. & Maish, T. (2017), "Extractive Text Summarization Using Word Vector Embedding," International Conference on Machine Learning and Data Science (MLDS), Noida, pp. 51-55. doi: 10.1109/MLDS.2017.12.

2. Akash, S., Avishek, Y. & Akshay, G. (2018), “Extractive Text Summarization using Neural Networks” at arxiv:1802.10137v1.

3. Chin-Yew Lin (2004), Rouge: A package for automatic evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop (pp. 74–81).

4. Efat, Ibrahim, & Humayun, K. (2013), “Automated Bangla text summarization by sentence scoring and ranking,” in Proceedings of International Conference on Informatics, Electronics & Vision (ICIEV), Dhaka, Bangladesh, 2013, pp. 1-5.

5. Kosrow, K. (2004), "Automatic Text Summarization with Neural Networks", *in Proceedings of the second International Conference on intelligent systems, IEEE* (pp. 40-44), Texas, USA.

6. Kamal, S. (2012), “Bengali text summarization by sentence extraction,” in *Proceedings of International Conference on Business and Information Management* (ICBIM-2012), Durgapur, India, 2012 (pp. 233-245).

7. Lajanugen & Honglak, L. (2015), “An efficient framework for learning sentence representations”, at arxiv: 1803.02893.

8. Lin, C., 2005. Recall-oriented understudy for gisting evaluation (rouge). *Retrieved August*, *20*, p.2005.

9. Mahmud, M.R., Afrin, M., Razzaque, M.A., Miller, E. and Iwashige, J., 2014, September. A rule based bengali stemmer. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 2750-2756). IEEE.Mahmud, M.R., Afrin, M., Razzaque, M.A., Miller, E. and Iwashige, J., 2014, September. A rule based bengali stemmer. In 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 2750-2756). IEEE.

10. Majharul, H., Surayia, P., & Zerina, B. (2017), “An Innovative Approach of Bangla Text Summarization by Introducing Pronoun Replacement and Improved Sentence Ranking”, J Inf Process Syst, Vol.13, No.4 (pp.752-777)

11. Mihalcea, R. and Tarau, P., 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404-411).

12. Rama, Bharagav, A. (2017) , "Text Summarization using Sentence Scoring Method", International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 04 , e-ISSN: 2395 -0056, p-ISSN: 2395-0072.

13. Tomas, M., Ilya, S. & Kai, C. (2013) , "Distributed Representations of Words and Phrases and their Compositionality", accepted to NIPS 2013.

14. Tas & Kiyani, F . (2017). A survey of automatic text summarization. PressAcademia Procedia , 5 (1) , 205-213 . DOI: 10.17261/Pressacademia.2017.591

15. Güneş, E. (2000). "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization". Available from <https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume22/erkan04a-html/erkan04a.html> [6 December 2019]

16. Kushal, C. (2011). "Text Summarization using Sentence Embeddings. Available from <https://medium.com/jatana/unsupervised-text-summarization-using-sentence-embeddings-adb15ce83db1> [6 December 2019].

17. Yonatan, H. (2010). "sentence representation with deep learning". Available from <https://blog.myyellowroad.com/unsupervised-sentence-representation-with-deep-learning-104b90079a93> [6 December 2019]