Enhancing Bangla text summarization in a monolingual setting

by

Muskan Ahmed
20301197

A.S.M Mahabub Siddiqui
20301040

Ayon Das
20301099

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Muskan

Muskan Ahmed

20301197

Mahabub

A.S.M Mahabub Siddiqui

20301040

Ayon

Ayon Das

20301099

Approval

The thesis titled "Enhancing Bangla text summarization in a monolingual setting" submitted by

1. Muskan Ahmed (20301197)
2. A.S.M Mahabub Siddiqui (20301040)
3. Ayon Das (20301099)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 17, 2024.

Examining Committee:

Supervisor:
(Member)

Farig Yousuf Sadeque

Farig Yousuf Sadeque

Associate Professor
Department of Computer Science and Engineering
School of Data and Sciences
BRAC University

Program Coordinator:
(Member)

Dr. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Associate Professor; Chairperson
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Abstract

Text summarization entails the automated generation of a short overview of a long text, such as an article, paper or collection of documents, while maintaining its main details. It aims to compress the size of the original text while maintaining its fundamental significance and expressing the key concepts. Our target is to enhance the precision and insightful summarization of monolingual Bengali writing. In a monolingual setting, to summarize Bangla text more precisely we will gather a dataset. The design of sophisticated models and algorithms will develop the text summarization of the Bengali language, thus aiding in the exploration of the distinct linguistic characteristics of Bengali. Bangla text Summarization is beneficial for numerous individuals, such as scholars, learners, teachers, reporters, creators, technology enthusiasts and language learners, who confront Bangla texts regularly. In short,those who work with Bangla language documents and want a concise synopsis of extensive materials would find it useful.

Keywords: Text Summarization, Summary, Bangla Text, Monolingual, Linguistic.

Table of Contents

List of Figures

Chapter 1

Introduction

Text summarization means rephrasing a long document into a condensed one with proper meaning. This process involves important details from the text and minimizing unnecessary duplication. Text summarization is a Natural Language Processing (NLP) process. NLP is the study of computer systems interacting with human language, specifically in the field of computational linguistics. NLP refers to a broad variety of activities associated with language (e.g. analyzing text, recognizing speech, translating languages, sentiment analysis etc). It allows computers to explain, produce, and respond to human language in a significant manner. There are two well-known methods for condensing text known as extractive and abstractive summarization. Extractive summarization chooses and extracts significant sentences straight from the original text to create a summary. On the other hand, Abstractive summarization entails producing a brief and logical summary by restating the information. Numerous methods for summarization have been studied for the extractive and abstractive techniques. The objective is to assist in effective information intake, simplifying the process for readers to quickly understand the primary message of a document. However, we will research Bangla text summarization. A variety of techniques have been devised to condense English texts. However, because of the complicated characteristics of the Bengali language, some limited attempts have been made. In this study, we will employ Abstractive summarization technique. Our objective is to improve the standard of Bangla summarization by employing enhanced preprocessing, fine-tuning, extensive datasets, or more advanced procedures.

Chapter 2

Research

2.1 Problem Statement

Automated text summarization is one of the most used and important tools in our life, whether it is very useful to understand the main context of a very large context within a very short time. Though English text summarization is quite rich with the use of its efficient summarization methods. But Bangla text summarization is not that kind of good position. To improve the position of Bangla text Summarization our work focus is to develop a quite good method which can give a satisfactory result to get a summary over large Bangla text. To work with this kind of problem some common challenges are faced such as repetition of same words multiple times, sometimes summary can not generate the main context of the input text where some most important informations are missed, length control with containing the main context of text etc. The main focus of our research is to build an effective method which can handle these problems as much as possible to generate a concise summary containing the main theme from a large document. By building this summarization method a wide range of people will be benefited who work with Bangla literature, students who try to get a summary from Bangla text or newspaper and lastly the mass people who use Bangla in their day to day life.

2.2 Research Objective

- The aim of the model is to develop a Bangla to Bangla text Summarization.

- It will be able to generate Bangla summaries from large Bangla text.

- The approach will allow users to quickly understand an extensive document by saving users time.

- Students, authors and those who deal with the Bangla language in their daily life will benefit from this method.

- We intend to release this system and publish our work after the research is complete.

Chapter 3

Literature Review

We looked into a few scientific articles that have been published recently and have a good number of citations in order to understand the scope of the research on this subject.
A paper by Kamal Sarkar[6] researched the use of keyphrases to textually summarize single Bangla and English documents.In order to create an extractive summary, this method first extracts a set of keyphrases from the document, utilizes those keyphrases to choose specific sentences from the document, and last, combines the chosen sentences. This method has three main steps for summarizing a document, which are: Preprocessing, keyphrase extraction and summary generation. The input document is separated into a group of sentences in the preprocessing step, and each sentence is assigned a unique sentence number. They have employed a keyphrase extraction module that has two simple processes: identifying potential keyphrases and ranking potential keyphrases. Here, a sequence of words free of punctuation and stop words is a candidate key phrase. They have also added some common verbs as stop words because keyphrases hardly contain common verbs. Two of the changes made to their system to make it Bangla-compatible are a Bangla stop wordlist and a Bangla stemmer. Their method of extracting candidate keyphrases includes two steps. First, potential keyphrases are extracted using stop words and punctuation as phrase boundaries. Next, longer phrases selected in the first stage are broken up into shorter phrases. For example, if one candidate keyphrase generated by the first step is “identify potential risk factors” then the second step will generate keyphrases “identify, potential, risk, factors, identify potential, potential risk, risk factors, identify potential risk, potential risk factors”. Potential keyphrases are ranked according to their weight before being chosen as the keyphrases. In this case, the weight of the potential keyphrase was determined using the PF (Phrase Frequency) and IDF (Inverse Document Frequency) metrics. Finally, their system for choosing sentences and creating summaries operates in two steps. Step 1 looks at the first n sentences in a document as potential summary sentences, ignores the remaining sentences, groups the potential summary sentences into groups where each group contains sentences that contain the same key concept, and then chooses the best sentence from each group to serve as the summary sentence. Step 2 is initiated if the proper summary length is not reached in step 1. In Step 2, each sentence that will be used in the summary is revised and ranked using the weights. The next step is to choose sentences one at a time from the ranking list; if the sentence hasn’t already been selected for the summary made in Step 1, it is added to the summary. Step 1 of the summarization technique described here is more stringent than step 2, which is only employed when step 1 is unable to provide the necessary length summary. They put their suggested keyphrase-based summarization strategy to the test

using two separate datasets: one for Bengali and one for English. For the evaluation of the English version, they have used DUC 2002 datasets. Their system has achieved better results than the baseline results of DUC in both cases of stopword removed and stopword not removed. The ROUGE-1 F-Score in case of stopword removed is 0.4308 [0.4220-0.4395] and The ROUGE-1 F-Score in case of stopword not removed is 0.4855 [0.4783 - 0.4925]. For the evaluation of the Bangla version, they have used documents that were summarized manually by humans. To measure the performance they compared their system to the existing Bengali summarizing systems. They had better results than the other systems. The F-Score of their approach is 0.4242 whereas the LEAD baseline F-Score is 0.4090.

Alvee Rahman et al.[13] suggested an extractive text summarization that simplifies Bangla text material using Fuzzy C-Means, TextRank, and aggregate sentence scoring techniques. From those three methods, they tried to find the most accurate method for Bangla text summarization. Their method has three steps for generating the summarization. Firstly, they pre-process the given text using tokenization, stopword removal and stemming. Here tokenization is used to divide each sentence into separate words. In the Stopwords removal process, it removes words that don't help in summarization, such as অবশ্য, এই, অথচ, কয়েক. The process of stemming is used to find the root word. For example, the root word of কাজের, কাজটি is কাজ. So, this process will stem the word কাজের or কাজটি to কাজ. TF-IDF Scoring, Sentence Length Scoring, Numerical value-based Scoring, Cue/Skeleton Word Scoring, Topic Sentence Scoring, and Sentence Position Scoring have all been employed in the sentence scoring procedure. The TF-IDF score, which stands for Term Frequency-Inverse Document Frequency, indicates how significant a certain word is throughout the entire text. They used Numeric value-based Sentence Scoring because numerical values significantly improve the summary. Cue/Skeleton Word Scoring is used by them because cue words like কারণ, যেহেতু, অতএব etc. can hold the gist of the whole text which is important for summarization. In their system, they compared other sentences in each given paragraph with the topic sentence, and the sentences that include the topic sentence's words are given higher priority. They used Position based scoring where the topic sentence and final sentence are given priority when reading through the paragraphs. Finally, for sentence extraction and summary generation, they used FCM, TextRank, and Aggregate Sentence Scoring algorithms. Fuzzy C-Means(FCM) is a clustering model that was applied in their system and was obtained using 2-dimensional data produced by the PCA model. Using the input data, the FCM model automatically determined the ideal number of centers. Afterward, iteratively determining which center the data points are closest to resulted in the creation of the clusters. The Fuzzy Partition Coefficient (FPC) value is used to determine the optimal amount of centers; the higher the value, the more accurate the selection of centers. A Textrank algorithm was used by them to compare the two models. Aggregate Sentence Scoring was used to provide a summary using features obtained through analysis of a certain text extract. They have used a dataset from an online repository where text and human-generated summaries are given. As texts for text summarizing, news pieces from several national daily newspapers have also been taken into consideration. Out of the three models they have used in their system the model Fuzzy C-Means has a higher F1 Score than the both TextRank and Aggregate Sentence Scoring models.

Alexander M. Rush, Sumit Chopra and Jason Weston[8] proposed an abstractive based sentence summarization that simplifies sentence summarization using a neural attention model. Their model is structurally very simple but can be trained with a huge amount of training-data. This model uses a problastic based neural language model and an encoder. To generate a summary they have used a generation algorithm decoder. Their problastic based neural language model was driven from a feed-forward NNLM(Neural Network Language Model) model. Their neural language model has four parameters (E;U;V;W). E is the word embedding matrix and U;V;W are weighted matrices. In case of encoders they have used three well known encodes.These are Bag-of-Words Encoder, Convolutional Encoder and Attention-Based Encoder. At first they used Bag-of-Words Encoder but it has some modeling issues. To overcome this issue they used a deep Convolutional Encoder. But this encoder improvement also has an issue of representing the whole input sentence. So they have modified their encoder and used an Attention-Based Encoder. To generate a summary they have used a Beam-Search decoder. Lastly to improve their model they have used extractive tuning, which has helped their model to find extractive word matches when needed. For training they have used an annotated Gigaword dataset which has more than 9.5 million news articles. For testing purposes with Gigaword they have also used DUC-2003 and DUC-2004 datasets. To compare their model result with other models they have used various baseline models. Those are IR, PREFIX, COMPRESS, W&L, TOPIARY and MOSES+. For the DUC-2003 and DUC-2004 datasets, their model ABS(Attention-Based Summarization) has over performed all the models but couldn't beat the MOSES+ in ROUGE-1, ROUGE-2 and ROUGE-L scores. But their model ABS+ did overpermed all the models with ROUGE-1 (28.18) , ROUGE-2(8.49) and ROUGE-L(23.81) scores. In the Gigaword dataset both their models ABS and ABS+ have overperformed all the other baseline models. In this case the ABS model has ROUGE-1 (30.88) , ROUGE-2(12.22) and ROUGE-L(27.77) scores.The ABS+ model has ROUGE-1 (30.88) , ROUGE-2(12.65) and ROUGE-L(28.34) scores.

The topic of Automatic Text Summarization is the main emphasis of the work by Mohamed Abdel Fattah et al.[3] Sentence position, sentence centrality, positive and negative keywords, sentence similarity to the title, sentence inclusion of name entity, sentence relative length, sentence inclusion of numerical data, Bushy path of the sentence, and aggregated similarity for each sentence have all been used in this trainable system to generate summaries. For sentence position they gave each sentence a ranking based on their position, taking into account a maximum of five positions. For instance, the first sentence in a paragraph receives a score of 5/5, followed by a score of 4/5 in the second phrase, and so on. Positive keywords are the key that commonly appears in the summary and Negative keywords are those that will probably not appear in the summary. Using sentence centrality they checked the similarity among all the sentences. They used sentence inclusion of name entity to check proper nouns in sentences. Typically, the sentence with the most proper nouns is the most significant and is most likely to appear in the summary. Which is also the same for numerical data. The feature sentence relative length is used to cut down the short sentences. The bushiness of a sentence is determined by examining its bushy path. High-bush sentences are interconnected with many other phrases, thus they share vocabulary and are more likely to discuss topics that are covered in many other sentences, which helps with summarizing. They have used aggregated similarity to calculate the importance of a sentence. Then, they trained genetic algorithm (GA) and mathematical regression (MR) models using the all-features score function to determine the best combination of feature weights. Since this is a trainable system for training GA and MR models they used 50 manually summarized English text documents(After feature extraction). For testing the models they used 100 English text documents(After feature extraction) which are different from the training text documents. To test their system they applied it to 100 religious articles. They assured that their system has overperformed the baseline approach.

The proposed system by Tahmid Hasan et al.[23] is ‘ Large-Scale Multilingual Abstractive Summarization for 44 Languages’ which is an abstractive text summarization. In this system, they provide XL-Sum, a vast and varied dataset made up of 1 million BBC article-summary pairings that have been professionally annotated and extracted using a series of carefully constructed heuristics. The dataset includes 44 languages that range in resource requirements from low to high, many of which lack a current public dataset. Regarding the number of samples gathered from a single source and the variety of languages included, to their knowledge, XL-Sum is the largest abstract summarizing dataset. They experiment with multilingual and low-resource summarization problems and fine-tune mT5 with XL-Sum. mT5 is the massively multilingual T5 text-to-text transformer model from Google. They carried out summarizing experiments in two different contexts: multilingual and low resource. Multilingual training involves using training data from various languages to train a single model. The multilingual model exceeded 11 ROUGE-2 scores across all languages. Although several of these languages are low-resource, the model nevertheless produced results that were competitive. Furthermore, they were the first to disclose the abstractive summarization benchmark for a variety of languages, including Bengali. To confirm that the dataset is versatile they train the model on individual languages. In comparison to all models trained on a single language, the multilingual model performed better. But the difference was less than 2 for R-2 scores. In the future, they plan to look into the usage of their dataset for cross-lingual summarizing and other summarization tasks.

Paper authored by Jingqing Zhang, Yao Zhao, Mohammad Saleh and Peter J. Liu[21] proposed a method with a new self-supervised objective which is a pre-trained large transformer - based encoder-decoder model named PEGASUS. In this model, the sentences which are important in the input document are masked and from those important sentences one output sentence is generated by combining them which is the desired output summary. This model is evaluated on 12 downstream summarization tasks such as emails, stories, news, science, instructions, legislative bills etc. They used two tokenization methods in this work. These two are Byte-pair-encoding algorithm and SentencePiece Unigram algorithm. This evaluation is analyzed from 32k to 256k vocabulary size. For pre-training purposes they use C4 which consists of text from 350M web-pages and HigeNews which is a dataset of 1.5B articles. Their proposed pre-training objective GSG is used in this work. Also BERT’s masked language model objective is used for comparison. GSG is basically a designed sequence-to-sequence self-supervised objective. In this step they considered 3 primary strategies. First one is Random, which is basically selecting m sentences randomly. The second one is Lead, which is selecting the first m sentences. The last one is Principle. In the last one, top m scored sentences are separated by their importance. Then they select those important sentences. By combining and comparing all these 3 strategies they select the most important m sentences. While comparing the results with GSG and BERT, it proved that Masked Language Model (MLM) does not improve downstream tasks. They used PEGASUS Base model to evaluate the choice of pre-training corpus while it was also proved that the Large PEGASUS model increased the capacity of larger hidden size with the same number of pre-training steps. The result of the Large PEGASUS model is quite good: it beat the previous result of 6 out of 12 datasets. They completed their full experiment into two steps. In the first step, the large PEGASUS model on both HugeNews and C4 is compared with the results of the base transformer. And in the second step, the large PEGASUS model is fine tuned and the results proved that they are at least as good as reference summaries. By analyzing their results, it proved that the summaries generated from this model were capable of achieving human performance on different datasets.

Paper authored by Nobel Dhar, Gaurob Saha, Prithwiraj Bhattacharjee, Avi Mallick, Md Saiful Islam[22] used a hybrid pointer generator network which is an abstractive model. This model is mainly focused on the repetition of words and lack of sufficient information in summary which is basically a major shortcoming of abstractive based text summarization. After completing dataset preprocessing Recurrent neural network (RNN) is implemented which works on sequentially data. As it works sequentially over every sentence it works quite good on text. At the first step encoder RNN reads the whole input text and after completing reading the decoder RNN starts to generate the summary. A probability distribution system named Attention distribution is used to find the highest probability distribution where to focus for the next word. Attention distribution creates a context vector with the weighted sum of encoder hidden states. After that generation probability distribution is calculated by combining the attention distribution and vocabulary distribution. The range of this score is between 0 to 1. Depending on the score it decides whether it generates a word or pointing towards any word from text. While creating words there must be a good chance for repetition of words. To solve this a mechanism named Coverage is used. In this process, the value of the attached coverage-vector with every word is changed after giving attention to the decoder. If any word is getting more attention this mechanism helps to find out that, as a result duplicity is reduced. To evaluate the accuracy of the system two rating matrices are used, one is quantitative and the other one is qualitative. To evaluate the quantitative evaluation they compare the Rouge value of those 4 dataset used in this model. Respectively the values of XL Sum, BANS 19k, pointer 19k and pointer 133k In Rouge-1 is 0.29, 0.59, 0.66, 0.67, In Rouge-2 the values are 0.12, 0.38, 0.41, 0.42 and lastly in Rouge-L 0.25, 0.49, 0.38, 0.41. To evaluate the qualitative result a survey over 20 people of 5 points has been done. Those results of the Bi-LSTM, BANS, Pointers on BANSData, Pointer on BANS-133 models are 2.75, 2.80, 3.13, 3.18.

Paper authored by Md. Nizam Uddin, Shakil Akter Khan[2] discussed some techniques for Bangla text summarization and some of them are implemented by them. The summarizer which is implemented by them is an extraction based and written in JAVA which is quite good for string processing. The methods they follow for summarizations are Location Method, Cue Method, Title, Term frequency and Numerical Data. In Location Method, which sentences are under headings and which position is quiet at beginning or end are considered as higher priority. In the Cue Method, which is specifically focused on the presence of pragmatic words in sentences. In this step, these sentences became more important. Also when we focused on the title of any passage the words which are used in the title got more importance than others. In the Title Method this process is focused to find out the words which are used in the title. In the frequency method the appearance of each word is counted and gets importance depending on the number of appearances. A word gets more importance which appears more in total. And lastly, the numerical data method is used to give more importance in those sentences with some numerical value than the sentences without numerical values. By following these methods this summarizer is implemented by taking the top 40 percent higher ranked sentences. To evaluate the summarizer two parameters are considered which are summary information and the size of the summary. By comparing the information of the original text a summary is considered how good the summary is. When two summaries contain the same information, the smaller one is prioritized. From a survey to measure the accuracy of the summarizer where size was ranked from 20 to 60 percent by giving 5 summaries of 10 articles, it found that 40 percent is the most acceptable size of the summarizer.

The Paper by Md. Iftekharul Alam Efat, Mohammad Ibrahim, Humayun Kayesh[5] proposed a method for Bangla text summarization which is Sentence Scoring and ranking. The total system is divided into three main segments, which are: preprocessing the input document, then scoring the sentence based on text extraction and lastly summarizing the text input based on sentence ranking. Before implementing the sentence scoring algorithm some preprocessing needs to be done. The preprocessing followed into three categories. Which are Tokenization, Stop words removal and Stemming. In the tokenization part, every sentence consists of more than one word and each word is considered as a token. In stop words removal segment, some Bangla words like তাই (So), এবং (And), অথবা (Or), কিন্তু (But) etc. are removed. Because these types of words are quite a good number in sentences but not of a heavy importance. Lastly in the Stemming part, all the words are transformed in their canonical forms. For example, যুদ্ধে, যুদ্ধের, যুদ্ধকে all are converted into its root form which is যুদ্ধ After completing the preprocessing unit the sentences of input are ranked based on their scores which would obtain from the four important features. Those four features are Frequency, Position Value, Cue words and Skeleton of the document. Frequency is measured from the occurrence of a word in the input document. Which words have more occurrences in the input those are counted as higher frequency and get more importance for summarization. Depending on the context of a document, the position of a sentence is quite important. Position value is computed by where a sentence appears in the input where it counts as a decreasing pattern, meaning the highest value is counted in the first sentence and then gradually it decreases. For summarization, cue words such as অবশেষে (Finally), সুতরাং (Therefore), অন্যদিকে (On the other hand) etc. are counted as an important factor. In those sentences if any cue words are used they get more importance. Lastly, in the measurement of the Skeleton of the document, those words are used in the title and headers get more importance than other words. Because by reading the heading of a passage it is quite easily understandable what is the main focus of the passage. After measuring all those four factors then the sentence score is counted which is a linear combination of these four factors. Based on the total score, sentences are ranked and providing the value of ‘X’ by a user the summary is made by combining those first ‘X’ number of sentences. This algorithm is already compared with human generated text summary and finally it found 83.57 percent average accuracy.

Paper authored by Ladda Suanmali, Naomie Salim and Mohammed Salem Binwahlan[4] implemented fuzzy method for text summarization which is an extraction approach. Before implementing this method, the dataset is preprocessed by four steps. Those are segmentation of sentences, tokenization, removing of stop words and stemming of words.

Detecting the boundary of sentences and separating texts from source into sentences is called segmentation of sentence. Tokenization is called separating text into words and considering them as token. Some stop words are used almost in every sentence, to get better results. These stop words are removed from the input. After that, in the last step all the suffixes and prefixes are removed from every word and converted to their root from. To determine the values of sentences, eight steps are followed. Those are Title features where the words which are used in the title are given more importance than others. To extract important information from sentences measuring sentence length is an important feature. By this process important information is extracted even from some short sentences. Term weight is calculated from the word frequency and by this score more frequent words get more priority. The position of the sentence is also important to measure the value of the sentence where the first sentence gets the most priority and it gradually decreases. By checking the similarity between sentences the value of sentence to sentence is measured and evaluated the most similar sentences in text. By finding the proper noun in sentences some important sentences can also be extracted. It is calculated by the ratio between the number of nouns used in a sentence and the length of the sentence. Another feature is, the top 10 most used words are used as thematic words and it is calculated by measuring the ratio of thematic words used in a sentence and max number of thematic words. Lastly, some sentences in text are always with some numeral data, these sentences got more priority than others. Then the fuzzy logic system is implemented in four steps. Where the fuzzifier is the first step and in this step inputs are translated into linguistic values. After that the inference engine refers this to the rule base and lastly the defuzzifier determines the final value by using membership function. After this process top 20 percent compression rate sentences are extracted. In the comparison to evaluate this process the highest correlation with human generated summary is 95 percent. Also when the fuzzy logic is compared with the baseline algorithm it performs well where the result of fuzzy logic is 0.49769 and baseline algorithm is 0.47002.

The paper by Sumya Akter, Aysa Siddika Asa, Md. Palash Uddin, Md. Delowar Hossain, Shikhor Kumer Roy and Masud Ibn Afjal[10] proposed a method for summarizing Bengali texts by extracting important information using the K-means Clustering Algorithm. Several methods have been created for summarizing English documents. But due to the complex nature of Bengali language, a few efforts have been attempted. This article presents a method for shortening text that chooses important sentences from one or more documents written in Bengali. The attempts made earlier were focused only on single documents or multiple documents but not both at the same time. However, this paper presents a method that can be used for both single document and multiple documents. They mentioned that extractive summarizers detect the most important sentences in the document and also remove the unneeded details. They utilize a technique of clustering sentences together to produce a summary from either one or multiple documents. In this method, the general pre-processing actions such as noise removal, tokenization, stop word removal, stemming are used. The Noise Removal process gets rid of unneeded data like header and footers from the document. Tokenization is the procedure of dividing the text into separate words. For Bengali text, words are divided by characters such as comma (,), full stop (।) etc. In the process of Stop Word Removal, typical function words like এবং (and) and কীভাবে (how) are taken out in order to make the information easier. For example, words such as বাংলাদেশ and বাংলাদেশর are subject to stemming and simplified to their basic form, which is বাংলাদেশ. This preprocessing step improves the effectiveness and precision of summarizing Bengali documents. Next, the total score of each sentence is calculated by adding together the words and their corresponding positions. TF*IDF is employed to determine the score of every word. If a particular sentence contains more unique words, it has relatively greater importance. If any cue word (e.g., মোটকথা, অবশেষে etc.) or skeleton word (e.g., title of any document) comes in any sentence, then the score of the sentence goes up by 1. Afterwards, the document is saved separately with the scores of the relevant sentences for future analysis. For multiple documents, all the steps (preprocessing, Scoring Process) are performed for every document and saved in one file. In this same file, they are later combined. So to continue with the processing, scores are arranged in a descending order. Lastly, the K-means clustering method is applied on the sorted list. In the K-means clustering algorithm, two centroids are assigned to the scores with the lowest and highest values. Sentences are then assigned to the nearest cluster based on closeness. Therefore, two centroids values are changed for upcoming iterations. This procedure is done repeatedly until two consecutive iterations yield the same outcome. Lastly, the top K sentences are chosen from each cluster to form the final summary. There are K statements that form 30 percent of all the statements in the combined original text. The authors determine that many experiments have been conducted to evaluate the suggested approach. Some are single documents and some are multiple documents. It lessens repetition and provides enhanced summarization. Additionally, the efficiency has improved when compared to different approaches. The time efficiency of the suggested approach is θ(n), which is quite beneficial. To enhance this technique, machine learning approach can be utilized in place of TF*IDF to predict sentence significance. In this case, parallel processing can be used for effectiveness, particularly when dealing with a substantial document. The main disadvantage is sometimes the arrangement of sentences is not synchronized. Therefore, to solve this issue,text cohesion analysis or even supervised learning models may be determined to determine the optimal order for sentences.The synchronization and coherence of the generated summaries may be evaluated using evaluation metrics, and the results can be utilized to make any necessary modifications. Moreover, Abstractive summarization can be utilized which creates brief sentences that may not be exact from the original text but convey the main idea of the content.

Another paper by Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çaglar G ulçehre and Bing Xiang[9] proposed a method called abstractive text summarization using Attentional EncoderDecoder Recurrent Neural Networks. In abstraction text summarization, a summary is created using different wording instead of just copying important sentences that already exist. The authors concentrated on abstractive text summarization since a majority of the current summarization techniques are extractive summarization that chooses sentences from the original text. However, this could potentially lead to summaries that are not coherent or similar to human summaries. They also aimed to enhance the quality and coherence of abstractive summaries and manage lengthier texts and effectively choose the most relevant information. For improving the precision and significance, They also focused on investigating the efficient utilization of pointers to indicate particular words or phrases in the original document. The authors utilized various models, with each one focusing on a particular weakness. Initially, they utilized the baseline model but in order to improve it, they implemented a technique known as the large Vocabulary Trick. Next, they employed Feature-rich Encoder. In order to understand the key concepts and significant aspects in the text, it is necessary to consider more than just the definitions of words. This includes incorporating language elements such as the roles of words, named items, and TF-IDF statistics. Furthermore, Managing unusual words in summarization, they utilized a switching generator/pointer model. Afterwards the Hierarchical Attention Model is employed to understand the significance of sentences alongside the understanding of wording for enhanced summarization.Finally, in order to decrease repeat phrases in generated summary, Temporal Attention Model is utilized. The models were originally trained using Gigaword Corpus but were tested on the DUC corpus. However, they also assessed their models on the CNN/Daily Mail corpus. After evaluating the datasets, the feats-lvt2k-2sent-ptr model (switching generator/pointer model) performs better than the rest in Gigaword Corpus. In CNN/Daily Mail corpus, the words-lvt2k-temp-att model (temporal attention) outperforms others and in DUC Corpus, the words-lvt5k-1sent model (a model with a larger LVT vocabulary size of 5k) performs better than the others. They asserted that The suggested models showed superior results in comparison to baseline models and previous state-of-the-art approaches, indicating a remarkable progress in abstractive summarization research. To enhance the method we could employ more sophisticated models such as transformers and more complex neural networks. Our emphasis should be on creating summaries that are grammatically accurate, logical, and simple to comprehend. Another possibility is to incorporate a function that highlights important details or main ideas in the summary, making it more informative.

The paper by Shusheng Xu, Xingxing Zhang, Yi Wu and Furu Wei[24] proposed a contrastive learning model for supervised abstractive text summarization. The proposed technique, SeqCo (Sequence Level Contrastive Learning), maximizes the similarity between representations of a document, its gold summary, and summaries generated in an adaptable manner. SeqCo promotes the encoding of important document details by representing them in the same vector space, leading to the production of more precise summaries.SeqCo modifies contrasting learning for the sequence-to-sequence learning scenario. In text summarization, the summary Y is a brief version of the input document X and they are expected to convey the identical message. Let Yˆ represent one example the model created from X. In essence, Yˆ should also have resemblance to both X and Y. Next, the transformer model (which includes an encoding transformer and a decoding transformer) is employed. The encoder Transformer changes the document into a sequence of hidden states, while the decoder Transformer predicts each component in the summary. SeqCo introduces an approach that emphasizes comparing education, where X, Y, and Yˆ are regarded as separate viewpoints of a mutual understanding. The aim is to enhance their resemblance while they are undergoing training. Contrastive learning is used to bring these perspectives closer together, ensuring that they represent the same topic. SeqCo consists of creating similarity computations between sequences, utilizing multi-head attention to contrast sequences, and executing a training loss that lessens the differences between these representations. The author used three different datasets to test their method. These are the CNN/DailyMail dataset, The New York Times dataset and XSum dataset. The authors provided details regarding the ROUGE scores for various models, emphasizing the superior performance of SeqCo in abstractive text summarization when compared to other models across various datasets. The SeqCo system is evaluated against extractive and abstractive summarization systems in the CNN/DailyMail dataset. SeqCo performs noticeably better than other models and also BART (When it comes to summarization duties, Bidirectional and Self-Regressive

Transformers is a well-known and established model in the area of natural language processing) in different setups that involve contrastive learning, showing the efficiency of the suggested method.In the dataset of the New York Times, SeqCo (x—yˆ) obtains significant enhancements. It achieved a score of 54.25 in ROUGE-1, followed by a score of 35.82 in ROUGE-2, and finally, it received a score of 50.24 in ROUGE-L. In the XSum dataset, the performance of SeqCo (x—y) is superior to all models that have been previously released, excluding Refsum and PEGASUS (which were trained on a vast dataset). It obtained a result of 45.65 in ROUGE-1, followed by a result of 22.41 in ROUGE-2, and ultimately, it was provided a score of 37.04 in ROUGE-L. The writer additionally demonstrated that their model obtains improved accuracy scores through Human evaluation. To improve this method, the training datasets can be expanded with more examples, this will help SeqCo learn better.

The article by Yang Liu and Mirella Lapata[16] suggested using pretrained BERT models (Bidirectional Encoder Representations from Transformers) as a means to enhance text summarization. BERT is highly effective for NLP tasks. They discuss how pretrained BERT in summarizing text and suggest a universal structure for both types of models - extractive and abstractive. They attempted to enhance the current level of automatic text summarization in order to ultimately enhance the caliber of the produced summaries. Additionally, their goal was to manage inputs consisting of multiple sentences and generate sentence representations of better quality. This approach is selected for producing summaries that contain more details and are logically organized, as compared to traditional models. It also addresses the restrictions in the original BERT model's maximum position length and modifies it for summarization. When BERT comprehends text, it analyzes separate words rather than sentences. However, for the purpose of summarizing, we aim to select complete sentences. To modify BERT for summarization, a BERTSUM model is created as a document-level encoder. To comprehend sentences, specific symbols ([CLS]) are appended at the beginning of every sentence. The writers additionally utilized section embeddings to assist BERT in recognizing the start of a new sentence. Furthermore, subsequent alterations in BERT result in enhanced comprehension of sentences. In BERT, the lower layers concentrate on individual sentences while the higher layers analyze multiple sentences and their connections. Then Position embeddings were incorporated to manage lengthier text. BERTSUM aids in extractive summarization by utilizing a unique vector that signifies each sentence. The vectors become special by using some math operation. After experimenting in various methods, they observed that the Transformer with 2 levels performed the topmost. The specific design is called BERTSUMEXT. In abstractive summarization, two parts are employed: an encoder (BERTSUM) and a decoder (Transformer). BERTSUM has extensive knowledge, whereas the transformer decoder begins with no prior understanding. To address this issue, a fine-tuning method is employed which is called BERTSUMABS. In order to improve the process, a two-stage fine-tuning approach is applied. This concept is named BERTSUMEXTABS.The authors conducted tests on three different datasets to demonstrate their accomplishments using BERTSUM. These datasets include the CNN/DailyMail dataset, The New York Times dataset, and the XSum dataset. The performance of the proposed method was evaluated using the ROUGE metric. In the CNN/DailyMail dataset, The New York Times dataset BERTSUMEXT, BERTSUMABS and BERTSUMEXTABS outperformed other previous methods except ORACLE(produces the best possible summary). In the XSUM dataset, the performance of the extractive model was not good, but BERTSUMABS and BERTSUMEXTABS demonstrated notable advancements compared to other methods. In order to enhance this approach, it is essential to comprehend the shortcomings of the model as this can direct the enhancements. Therefore, examining the model inaccuracies and fine-tuning the model to correct particular kinds of errors. Moreover, trying out various attention mechanisms can effectively capture superior significant details and connections within the document.

The research article by Li Wang, Junlin Yao, Yunzhe Tao, Li Zhong, Wei Liu, Qiang Du[14] suggested a advanced machine learning framework known as Convolutional Sequence-to-Sequence (ConvS2S) and self-critical sequence training (SCST) for improvement. The objective is to create informative and accurate summaries while handling difficulties like choosing significant information, and producing readable summaries for humans. The suggested framework incorporates joint attention and biased probability generation mechanism to enhance consistency and variety in summaries. The suggested model employs a convolutional architecture with input words and topics, a joint multi-step attention mechanism, and a biased generation structure. In order to improve effectiveness, it employs reinforcement learning, particularly self-evaluative sequence training (SCST). The structure employs the Convolutional Sequence-to-Sequence (ConvS2S) concept, integrating convolutional elements to produce representations for both word and topic levels. The arrangement of data in a specific order can be maintained by using position embeddings. Convolutional layers use a filter of size k to process the input elements, and the resultant output is later passed to the Gated Linear Unit (GLU) function for further computation. Multi-stage Attention is employed to ascertain the influence of input components on the decoder's condition via attention weights. The topic model is used to extract hidden information from documents, and the multi-step attention-based ConvS2S model includes a topic-aware technique to include prior knowledge for text summarization. Pre-trained Latent Dirichlet Allocation (LDA) models are used to generate topic representations, which capture hidden comprehension of documents. The collective focus procedure enhances the convolutional configuration by integrating the individual's data during decoding. Partiality in probability generation employs a variety of consequences for sought-after components, considering the outcomes from the decoder at both the word and topic levels. Furthermore, they used a reinforcement learning method called self-critical sequence training (SCST) to reduce the influence of biased text summarization. This was done by maximizing the ROUGE metric, which cannot be differentiated, and generating sequences using both a greedy method and a sampling method to enhance the consistency of the training and testing processes. The model has six convolutional layers, a scaling factor of 0.99, 256-dimensional embeddings, and a 0.25 learning rate. The experiment was performed over Gigaword, DUC-2004, and LCSTS datasets. In the Gigaword Corpus, the ROGUE-1 score of Reinforced-Topic-ConvS2S is 36.92, the ROUGE-2 score is 18.29, and the ROUGE-L score is 34.58. The DUC-2004 dataset exhibits ROGUE-1 with a score of 31.15, ROGUE-2 with a score of 10.85, and ROGUE-L with a score of 27.68. In the LCSTS dataset, the ROGUE-1 scores are 39.93/45.12, the ROUGE-2 scores are 21.58/33.08, and the ROUGE-L score is 37.92/42.68. The suggested framework demonstrates the greatest ROUGE scores when compared to alternative models. To improve this model, Trying out different levels of convolutional layers, kernel sizes, and hidden layer sizes might help.

The paper by S. M. Afif Ibne Hayat, Avishek Das, and Mohammed Moshiul Hoque[25] proposed an automated abstractive summarization system using transformer based models for Bengali text in the paper. The goal of this work is to bridge the current gap between the more resourceful high resource languages and less resourceful Bengali in the domain of summarization. In the paper, five (B-T5, B-T5-Base, mT5-Small, mT5 Base, and mBART-50) different transformer based approaches were evaluated and they found that the Bangla-T5 is the best performing model. The proposed model includes the text-to-text-transfer Transformer (T5) model. For the models, they used two publicly available datasets, namely BANS and XLSum. BBC Bangla has 10,126 document summary pairs in XLSum and BANS has 19,000 pairs. The transformer models are fine tuned on the XLSum dataset and then reevaluated in the combined dataset. The T5 model is an encoder-decoder architecture. Self attention and feed forward layers in the encoder take input and produce attention vectors. These vectors are used by the decoder to get outputs and repeat this until completion. The output is fed to a linear layer then softmax and then gets the final probabilities. It contains the Bengali text normalization and tokenization using SentencePiece to convert numeric text input into something that a transformer understands. T5 base models along with their multilingual variant are all pre trained on the C4 corpus (mC4). BART is a seq2seq model trained as a denoising autoencoder. It means the model takes an input text sequence (which can be noisy, as a few words might be missing) and returns the corrected version of it. Because of this training style BART learns how to generate coherent text that is close to natural human language, and is thus effective for text summarisation and other natural language processing tasks. The weighted ROUGE F-measure scores were used to compare the performance of the models and the Bangla-T5 (B-T5) model achieved the best ROUGE scores across transformer based models on the XLSum dataset (27.59 ROUGE 1, 10.78 ROUGE 2, and 23.7 ROUGE L). Additionally, the B-T5 model yielded the highest ROUGE scores on the merged dataset (33.58 ROUGE-1, 13.83 ROUGE-2 and 26.24 ROUGE-L). On the XLSum dataset, B-T5 achieved ROUGE-2 score of 10.78, which was a 57.17% improvement over mT5-Base(6.86). For the merged dataset, The B-T5 demonstrated a 52.31% improvement in ROUGE-2 (13.83) over mT5-small (9.08). Overall, it seems that B-T5 outperforms multilingual models, especially when trained on language specific datasets. Training with the merged dataset led to an average 35.3% improvement in ROUGE-2 scores for the models, with B-T5 showing the most consistent performance. Nevertheless, B-T5 outperformed other models in both datasets. However, this method can be further improved by enlarging datasets to learn more. More high quality text-summary pairs, and from multiple domains (e.g., news, medical, academic) allow the model to be better trained. Making sure the dataset is clean and has a proper associated summary will reduce any ambiguities in the data which result in unpredictability, causing issues for the model.

The paper by Yinhan Liu et al [19] presents a simple approach to improve the quality of machine translations (MT) in several languages, including low-resource languages by making use of multilingual pre-training. In this work, the authors present mBART (Multilingual BART) a sequence-to-sequence denoising auto-encoder, pretraining on multiple languages. To do this, the team improved supervised and unsupervised machine translation quality by introducing pre-training models which can share multilingual knowledge in the nature of cross-aligning sentences across languages directly also for language pairs that have no parallel data. As the researchers explain, their work aims to overcome drawbacks in current MT systems that either pre-train portions of model components or concentrate on high resource languages like English. They pre-train mBART on CC25 a subset of 25 languages and correctly corrupted inputs using the noise functions such as span masking, sentence permutation to reconstruct text. They further fine-tuned the new model on low-resource and high-resource translation pairs, validating it in both a sentence-level test and document-level test. The mBART model led to substantial enhancement. It even gained 12 BLEU points over a few language pairs with lower resources. In addition, the model showed promising zero-shot transfer translating between languages it had not been trained on. The experiments also demonstrated improvements in document-level translation, as the proposed method excelled over baseline systems and delivered state-of-the-art results for a number of downstream tasks involving low-resource languages.

The paper by Kira Sam and Raja Vavekanand [30] focuses on creating a gigantic LLM called Llama 3.1 having over 405 billion parameters. The main objective was to build a model which is efficient, scalable and highly customizable that could work well on multilingual and complex reasoning tasks. They also wanted to address the model's handling of long context windows up to 128K tokens, challenging it with tasks like summarization, question answering and content generation.The model was trained on a large dataset of 15 trillion tokens drawn all over the net from websites, books or research articles. Utilizing more than 16,000 H100 GPUs for massive parallel training over advanced and flexible network configurations. They concentrated on refining the architecture of their model by a standard dense Transformer model with enhanced training stability by avoiding the complexity of mixture-of-experts models. Then they fine-tuned the model using techniques like supervised fine-tuning (SFT) and rejection sampling. In many benchmarks, such as multilingual translation and coding or reflection tasks, Llama 3.1 could outperform its predecessor Llama 2 andeven a model like GPT-4. They achieved state-of-the-art performance across more than 150 benchmarks and displayed strong zero-shot capabilities. This model is highly scalable and efficient, able to facilitate a variety of use cases across many industries ranging from content creation to automated customer support.

Chapter 4

Dataset Analysis

Dataset analysis is very important in the field of machine learning , data science and especially in natural language processing. It gives us various angles and aspects of the data which helps us with the research work.

Dataset analysis firstly helps us to understand the data clearly. By this we can clearly know the data distribution, the variables of the dataset and more importantly the relationship between the data. It also gives us the information if the dataset has any error or inconsistency. Which helps us to insure the quality of the dataset. Moreover, analyzing the data set can be crucial in case of feature selection. We can easily understand which features are important for our work by dataset analysis. Dataset analysis is very crucial in case of data pre-processing and model selection. Lastly, exploratory data analysis gives us the insight of data. Which can be crucial in discovering patterns in data. These patterns and insights of data can be helpful in the decision making process of the research.

4.1 Data Collection

Our research is based on Bangla text summarization. In the Neural Language processing field Bangla language is a low resource language.For Bangla language Creating or collecting dataset is one of the tough tasks. For our research we tried to collect data from open source resources.

For our research we have collected a dataset from BUET CSE NLP Group. XL- Sum, a vast and varied dataset made up of 1 million BBC article-summary pairings that have been professionally annotated and extracted using a series of carefully constructed heuristics. This data contains 44 languages where there are low-resource languages like Amharic, Igbo, Somali and Bangla to high-resource languages like English, Hindi and Russian. We have collected the Bangla portion of the dataset which will help us in our Bangla text summarization process.

The dataset we collected has only BBC article-summary pairings. So we tried to extend the dataset and create a new dataset containing data not only from BBC but also from other sources like The Daily Star, The Ittefaq and Prothom Alo. We tried to include different types of domains. But we only got article-summary pairings data from The Daily Star. We couldn't fetch data from The Ittefaq because it was restricted and from Prothom Alo because they don't have article-summary pairings data. The quality and robustness of a summarization model can be greatly improved by using different domains. Different writing styles, different content structures and different topics in datasets from different sources help explore models differently and the more models can generalize over wider spreads of inputs.

“id”:“news-48103267”,**“url”:** “https://www.bbc.com/bengali/news-48103267”, **“title”:** বিশ্বকাপ ক্রিকেট ২০১৯: বিতর্কের মুখে বাংলাদেশের জার্সিতে ’পরিবর্তনের আবেদন’ করেছে বিসিবি, **“summary”:** প্রবল সমালোচনার মুখে বাংলাদেশ দলের বিশ্বকাপ জার্সি পরিবর্তনের সিদ্ধান্ত নিয়েছে বিসিবি। তবে এজন্য এখন আইসিসির নিয়মের মধ্যে দিয়ে যেতে হবে বোর্ডকে। **“text”:** বিশ্বকাপের ফটোসেশনে বাংলাদেশ দল সোমবার বেশ আয়োজন করেই বিশ্বকাপ ক্রিকেট ২০১৯-এর জন্য বাংলাদেশ জাতীয় দলের জার্সি উন্মোচন করে বাংলাদেশ ক্রিকেট বোর্ড -বিসিবি। একইসাথে মিরপুর শেরে বাংলা স্টেডিয়ামে অনুষ্ঠিত হয় বিশ্বকাপগামী দলের অফিসিয়াল ফটোসেশনও। আর এই ছবি গণমাধ্যম ও সামাজিক যোগাযোগ মাধ্যমে ছড়িয়ে পড়তেই শুরু হয় নানা আলোচনা-সমালোচনা। জার্সির ডিজাইন ও রং নিয়ে প্রবল সমালোচনার মুখে পড়ে ক্রিকেট বোর্ড। বিশ্বকাপ জার্সি নিয়ে সামাজিক যোগাযোগ মাধ্যমে সমালোচনার ঝড় উঠে বিশ্বকাপ সামনে রেখে এবার হোম ও অ্যাওয়ে ভিত্তিতে দুটো আলাদা জার্সি করেছে বিসিবি। যার একটি সবুজ ও অন্যটি লাল। তবে যে সবুজ জার্সি পরে ফটোসেশন করেন মাশরাফি-তামিমরা সেটাতে কোন লালের ছোঁয়া না থাকাতেই আপত্তি তৈরি হয় অনেকের। এমনকি সামাজিক মাধ্যমগুলোতে হ্যাশট্যাগ দিয়ে জার্সি বদলের কথাও আসতে থাকে। পাকিস্তানের যে জার্সির সাথে মিল খুঁজে পাচ্ছেন ভক্তরা তবে বিসিবি বলছে, তাদের ডিজাইনে শুরুতে লাল রং রাখা হয়েছিল। "জার্সিতে কিন্তু শুরুতে লাল রং ছিল, আমরা বাংলাদেশ ও ক্রিকেটারের নামটা লাল রঙে লিখেছিলাম। কিন্তু আইসিসি আমাদের বলে সেটা সাদা রঙে দিতে,’’-জানাচ্ছিলেন বিসিবির প্রধান নির্বাহী নিজামউদ্দিন চৌধুরি। আইসিসি তাদের ফেসবুক ও টুইটারের কাভার ফটো করেছে বাংলাদেশ দলের এই ছবি দিয়ে। সব দেশকেই তাদের বিশ্বকাপ জার্সির জন্য আইসিসির অনুমোদন নিতে হয়। এবার তাই পরিবর্তনের ক্ষেত্রেও আইসিসির কাছেই আবেদন করতে হবে বিসিবিকে। যদিও এরইমধ্যে এই জার্সির ছবি আইসিসি সবখানে ব্যবহার শুরু করেছে। "আমাদের এখন আইসিসির অ্যামেন্ডমেন্ট অনুযায়ী যেতে হবে। এই মুহূর্তে তাই বলতে পারছি না কেমন পরিবর্তন, তবে একটা মাইনর চেঞ্জ হবে,’’-বলছিলেন নিজামউদ্দিন চৌধুরি। জার্সি নিয়ে সবচেয়ে বড় সমালোচনাটা হল পাকিস্তানের সঙ্গে এর মিল। ফেসবুকে দুই দলের জার্সি মিলিয়ে বিভিন্ন ছবিও ছড়িয়ে পড়ে। অনেকেই এই জার্সিকে পাকিস্তান বা আয়ারল্যান্ডের মতো বলে মন্তব্য করছেন তবে এটাকে ’নিতান্ত কাকতালীয়’ বলছেন বিসিবির মিডিয়া কমিটির চেয়ারম্যান। "পাকিস্তানের কাছ থেকে কেন কপি করবো আমরা, বরং ওরাই আরো আমাদের কাছ থেকে কপি করবে। দেখুন আমাদের জার্সি সবসময় লাল-সবুজ থিমের উপর করা হয়।" "এর সাথে কমলা বা হলুদ রংও কিন্তু যুক্ত হয়। এর আগেও হালকা সবুজ বা লাল রংয়ের জার্সি পরে বাংলাদেশ খেলেছে।’’ অধিনায়ক মাশরাফিকে নতুন জার্সি তুলে দিচ্ছেন বিসিবি সভাপতি তাহলে কি নেতিবাচক সমালোচনার কারণেই জার্সি পরিবর্তনের সিদ্ধান্ত নিতে হল বিসিবিকে? - এ প্রশ্নের উত্তরে এই দুই কর্মকর্তা বললেন "অনেকটা সেরকমই"।’ তবে আপাতত আইসিসির অনুমোদনের অপেক্ষায় থাকা ছাড়া আর উপায় নেই বাংলাদেশ ক্রিকেট বোর্ডের। বিবিসি বাংলার অন্যান্য খবর: জার্সি বিতর্ক: পাকিস্তান দলের সঙ্গে কতটা মিল বাংলাদেশের রাজধানী কি ঢাকার বাইরে নিতে হবে?

The dataset is stored in JSONL format. The dataset contains the following fields: id, url, title, summary, and text. Among these, the most important fields for our research are the summary and text, as these are key for performing intrinsic evaluations of the summarization models. This table serves as a demo of how our dataset is structured and what kind of information is stored in each field.

4.2 Data Pre-Processing

Data pre-processing is an important part of Natural Language Processing (NLP) which involves cleaning, preparing raw and removing unwanted data for model training. If the dataset is pre-processed properly it will lead to better model accuracy. So we have pre-processed our dataset which we have fetched from The Daily Star. Firstly, we have seen if there are any null values and removed them because it will negatively impact the summarization performance. Then duplicated data is discovered and discarded to prevent redundancy. Moreover, the texts that are not twice the length of the corresponding summary are identified and removed.

4.3 Exploratory data analysis for the XL-sum dataset

4.3.1 Dataset Overview

The dataset has a total of 10126 samples.The dataset is splitted into 8:1:1 split, where 80% of the samples are used for training the model. 10% is for validation and another 10% is for testing the model. So, we have 8102 training samples, 1012 testing samples and 1012 validation samples.

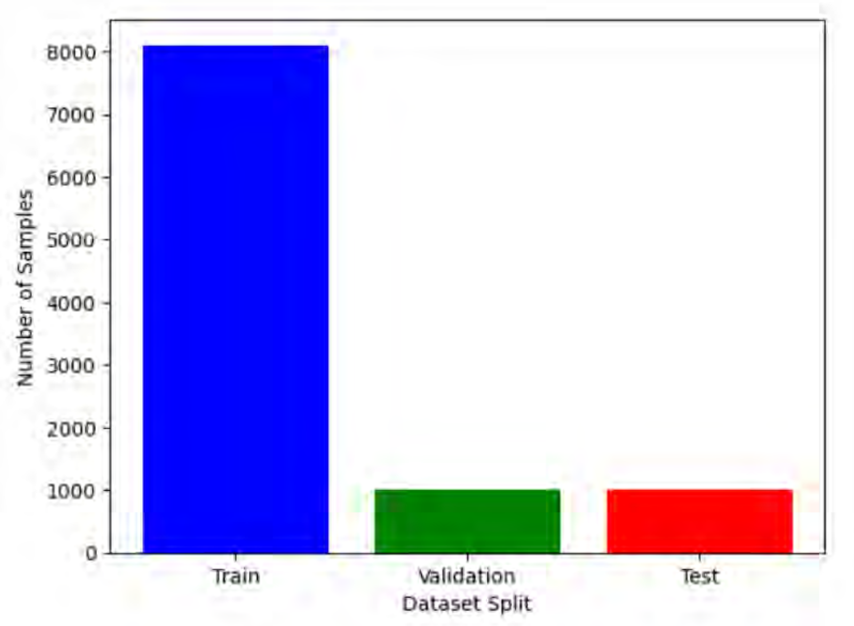


Figure 4.1: Dataset Overview

4.3.2 Text Length Distribution

In our dataset every data has three sections: title, text and summary. This histogram shows the text length distribution of the dataset. The X-axis represents the length of the texts in words. Which is labeled as “Word Count”. The Y-axis represents how many times a text of a certain length occurs in the dataset. This axis is labeled as “Frequency”. This histogram has a right skewed distribution. Which means most of the samples of the dataset have lower word count. From the histogram it is visible that the highest number of samples has 400-600 words. Also most of the samples have word count between 100 to 1500. Very few samples have word count over 2000 words. Lastly the average text length is 601.72 words.

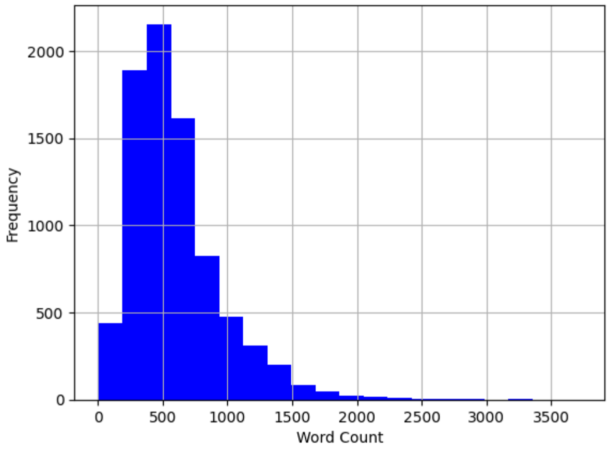


Figure 4.2: Text Length Distribution

4.3.3 Summary Length Distribution

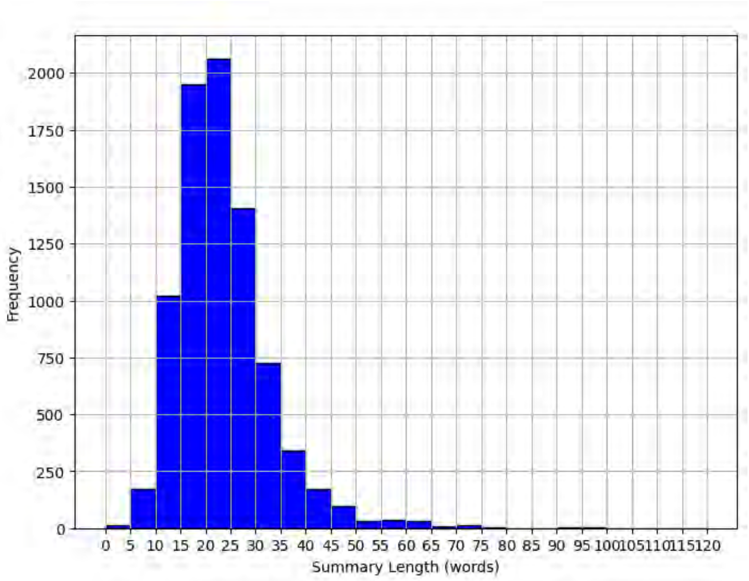


Figure 4.3: Summary Length Distribution

This histogram shows the text length distribution of the dataset. The X-axis represents the length of the sumarries in words. Which is labeled as “Summary Length (words)”. The Y-axis represents how many times a summary of a certain length occurs in the dataset. This axis is labeled as “Frequency”. his histogram also has a right skewed distribution. Which means most of the samples of the dataset have lower word count. In this histogram the summary length distributions of the training data can be seen. Here, the highest number of samples falls between 20-25 words. Most of the summary lengths are between 5 to 50 words. Very few summaries in training data are above 50 words. In the training dataset Average Summary Length is 23.04 words.

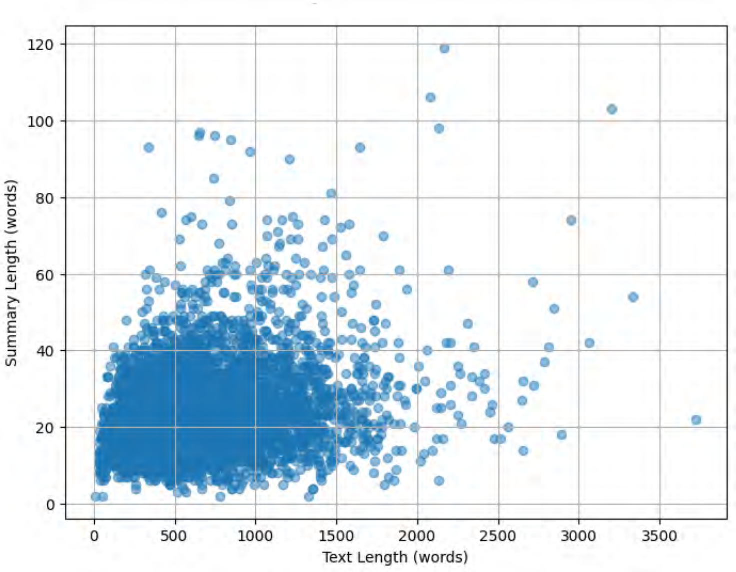


Figure 4.4: Text length vs Summary length

4.3.4 Text length vs Summary length

Here the Text length vs Summary length scatter plot is visible. The X-axis represents the length of the text in words. Which is labeled as “Text Length (words)”. The Y-axis represents the summary length in words. This axis is labeled as “Summary Length (words)”. From the scatter plot it is clearly visible that most of the texts have a summary between 10-40 words. Very few texts have summaries above 50 words. Also, when the text length increases the summary length also increases, but not so intensively. There are many texts with the length between 2000-3500 words but their summaries are not that long.

4.3.5 Uni-gram Analysis

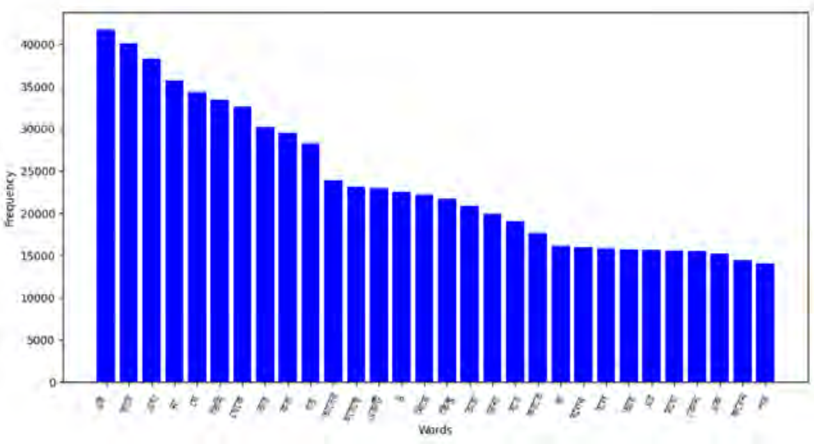


Figure 4.5: Uni-gram Analysis

This bar chart represents the frequency of unigram,which is the frequency of single words in the dataset. From the uni-gram analysis we can see that words like এই, করে, এবং, না is so frequent. The most frequent word in the training data is এই. Uni-gram analysis is so valuable in text summarization. It helps to identify the commonly or frequently occurring words. Frequent words in a dataset can be important for the summarization. Also, uni-gram analysis can help in data pre-processing. By uni-gram analysis the less important but commonly occurring words can be founded and can be added as stopwords.

4.3.6 Bi-gram Analysis

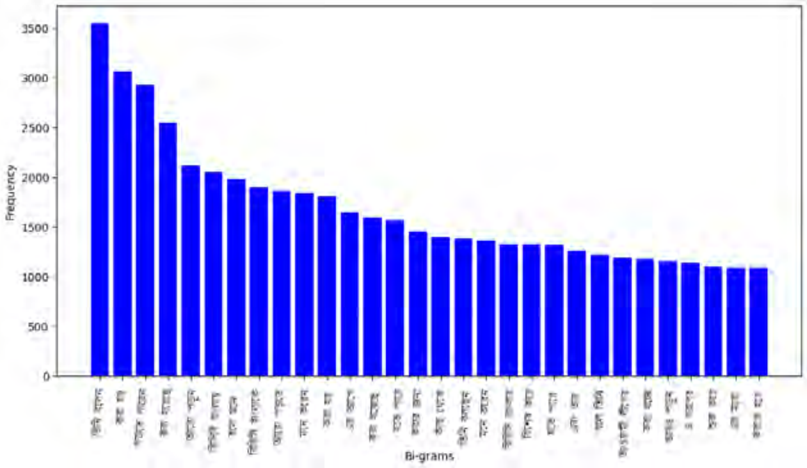


Figure 4.6: Bigram Analysis

This bar chart shows the frequency of Bi-grams, which is the frequency of two worded phrases in the dataset. Here the X-axis represents the common Bi-grams and the Y-axis represents the frequency of those common Bi-grams The most frequent phrase in Bi-gram analysis is তিনি বলেন. Phrases Like করা হয়, করা হয়েছে, পড়তে পারেন আরো পড়ুন is frequent in the train data. পড়তে পারেন, আরো পড়ুন such phrases are frequent in this data set because this dataset in created from the BBC news articles. For coherent summary generation understanding the context of the original text is necessary. Bi-gram analysis helps to find the commonly used phrases, which can help to capture context of a sentence. Moreover, Bi-gram analysis helps to resolve ambiguity. Some words can have different meanings in different contexts. Bi-gram analysis is very helpful in those cases.

4.3.7 Tri-gram Analysis

This bar chart shows the frequency of Tri-grams, which is the frequency of “sequence of three words ”. Here the X-axis represents the common Tri-grams and the Y-axis has the frequency of those common Tri-grams. From the Tri-gram analysis it is clearly visible that the phrase আরো পড়তে পারেন has the highest frequency. Phrases like বিবিসি বাংলায় আরো, বাংলার অন্যান্য খবর, বাংলায় আরো পড়তে are frequent in the dataset. Those phrases are frequent because this dataset is created from the BBC news articles. From Tri-gram analysis we can see which words tend to come together in many sentences. Also, tri-gram carries

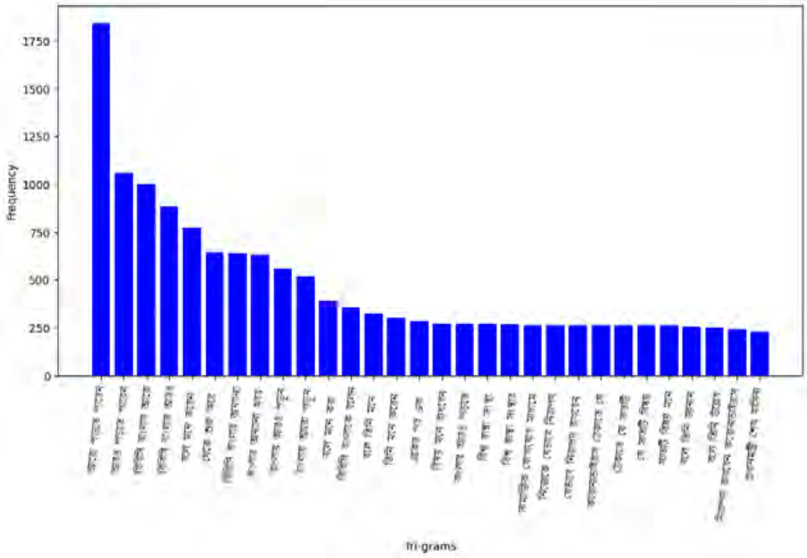


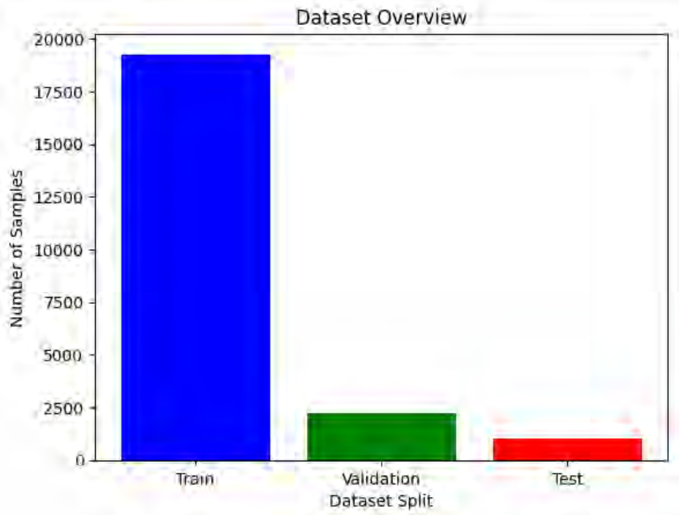
Figure 4.7: Trigram Analysis

more sequence and meaning which can be so helpful in removing ambiguity and capturing context of a text.

4.4 Exploratory data analysis for the merged dataset

4.4.1 Dataset Overview

The dataset has a total of 22531 samples.The dataset is splitted into an 8.55:1:0.45 split, where 85.5% of the samples are used for training the model. 10% is for validation and another 4.5% is for testing the model. So, we have 19266 training samples, 1012 testing samples and 2253 validation samples. We only extended the training data and the validation data in our extended dataset.



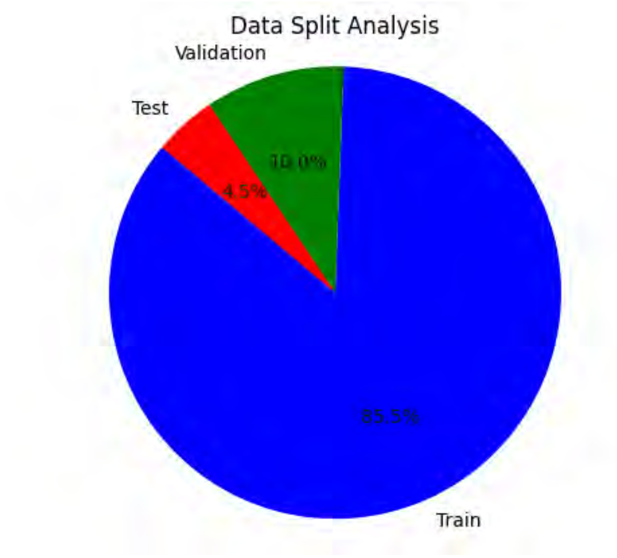


Figure 4.8: Dataset Overview

4.4.2 Text Length Distribution:

The X-axis represents the length of the texts in words. Which is labeled as "Word Count". The Y-axis represents how many times a text of a certain length occurs in the dataset. This axis is labeled as "Frequency". This histogram has a right skewed distribution. Which means most of the samples of the dataset have lower word count. From the histogram it is visible that the highest number of samples has 250-500 words. Also most of the samples have word count between 100 to 1500. Very few samples have word count over 2000 words. Lastly, the average text length is 542.25 words.

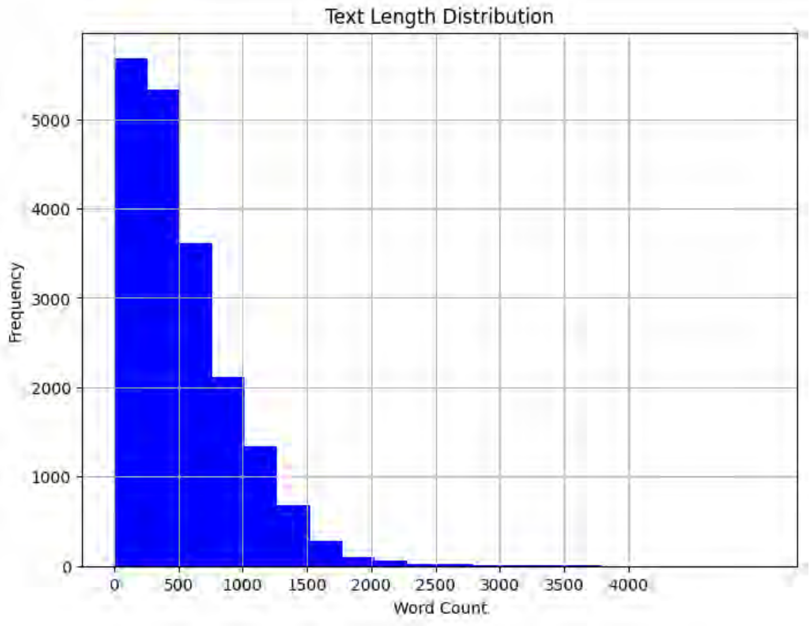


Figure 4.9: Text Length Overview

4.4.3 Summary Length Distribution:

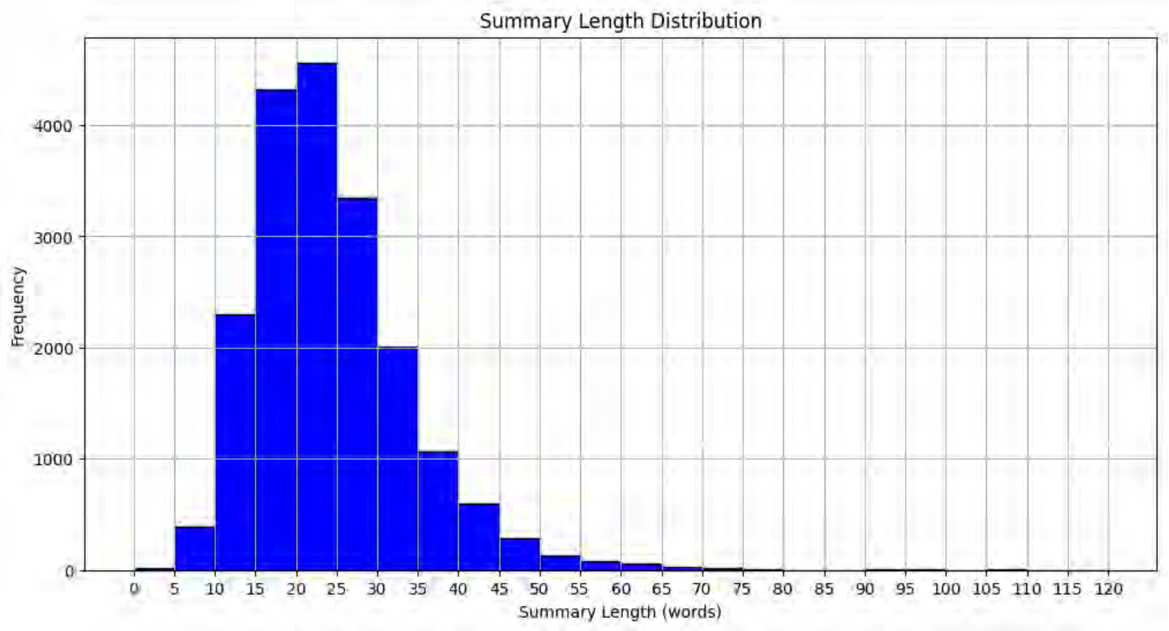


Figure 4.10: Summary Length Overview

This histogram shows the summary length distribution of the dataset. The X-axis represents the length of the sumarries in words. Which is labeled as "Summary Length (words)". The Y-axis represents how many times a summary of a certain length occurs in the dataset. This axis is labeled as "Frequency".This histogram also has a right skewed distribution. Which means most of the samples of the dataset have lower word count. In this histogram the summary length distributions of the training data can be seen. Here, the highest number of samples falls between 20-25 words. Most of the summary lengths are between 5 to 55 words. Very few summaries in training data are above 60 words. In the training dataset Average Summary Length is 23.82 words.

4.4.4 Text length vs Summary length:

Here the Text length vs Summary length scatter plot is visible. The X-axis represents the length of the text in words. Which is labeled as "Text Length (words)". The Y-axis represents the summary length in words. This axis is labeled as "Summary Length (words)".From the scatter plot it is clearly visible that most of the texts have a summary between 10-60 words. Very few texts have summaries above 60 words. Also, when the text length increases the summary length also increases, but not so intensively. There are many texts with the length between 2000-3000 words but their summaries are not that long.

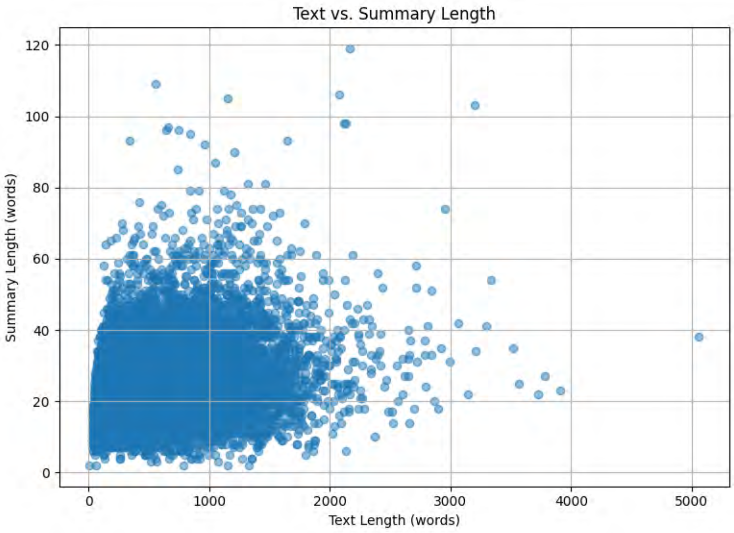


Figure 4.11: Text Length And Summary Length

4.4.5 Uni-gram Analysis:

This bar chart represents the frequency of unigram,which is the frequency of single words in the dataset. From the uni-gram analysis we can see that words like করে , এই , এবং , না is so frequent. The most frequent word in the training data is করে. This word has occurred more than 8000 times in the dataset.

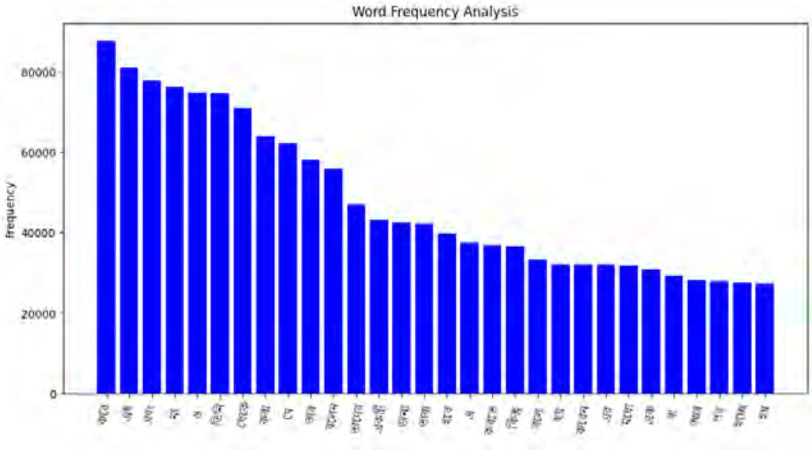


Figure 4.12: Uni-gram Analysis

4.4.6 Bi-gram Analysis:

This bar chart shows the frequency of Bi-grams, which is the frequency of two worded phrases in the dataset. Here the X-axis represents the common Bi-grams and the Y-axis represents the frequency of those common Bi-grams The most frequent phrase in Bi-gram analysis isতিনি বলেন. This Bi-gram has occurred more than 10000 times in the dataset. Phrases Like ডেইলি স্টারকে , বিবিসি বাংলাকে, করা হয়, করা হয়েছে, পড়তে পারেন, আরো পড়ুন is frequent in the train data. ডেইলি স্টারকে , বিবিসি বাংলাকে, পড়তে পারেন, আরো পড়ুন , such phrases are frequent in this data set because this dataset in created from the BBC and The Daily Star news articles.

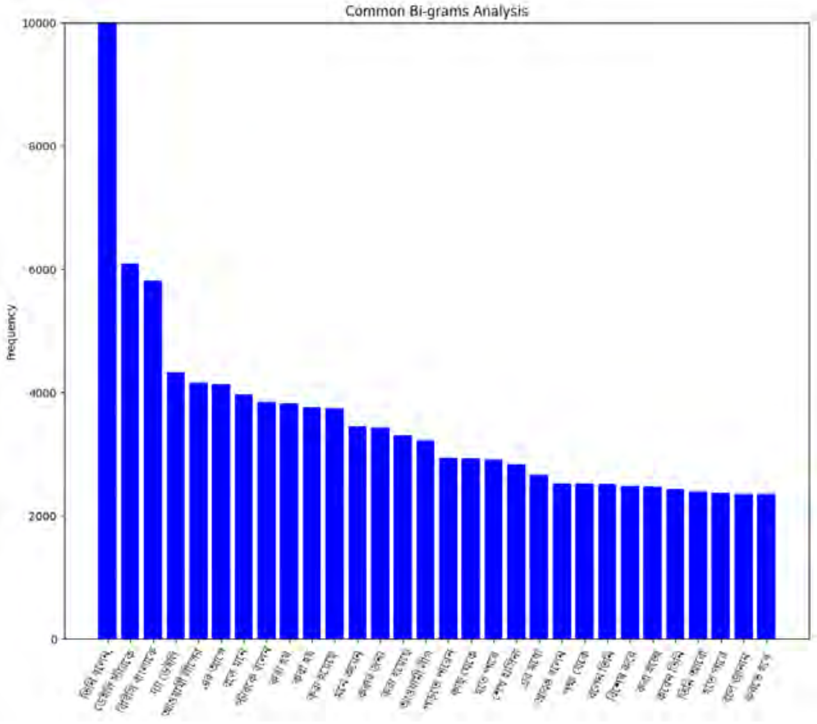


Figure 4.13: Bi-gram Analysis

4.4.7 Tri-gram Analysis:

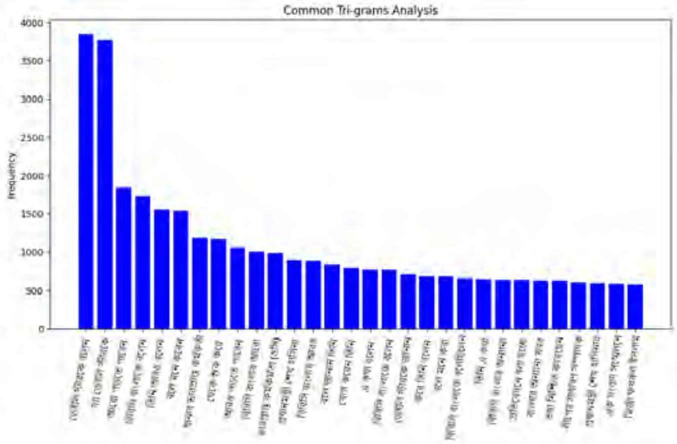


Figure 4.14: Tri-gram Analysis

This bar chart shows the frequency of Bi-grams, which is the frequency of “sequence of three words ”. Here the X-axis represents the common Trigrams and the Y-axis has the frequency of those common Tri-grams. From the Tri-gram analysis it is clearly visible that the phrase ডেইলি স্টারকে বলেন has the highest frequency. This Tri-gram has occurred more than 3500 times in the dataset. Phrases like ডেইলি স্টারকে বলেন, বিবিসি বাংলায় আরো, বাংলার অন্যান্য খবর, বিবিসি বাংলাকে বলেন are frequent in the dataset. Those phrases are frequent because this dataset is created from the BBC and The Daily Star newspaper.

4.5 Dataset Evaluation:

In natural language processing (NLP) dataset evaluation is important because considering the fact that the quality of the dataset is directly related to the performance of the model. Developing meaningful statistical models is impossible without having high quality data. In other words, the models will pick up the right task without learning irrelevant patterns or biases. Due to the unique challenge that the task poses in the context of summarizing, dataset evaluation becomes particularly important. In the merged dataset, we used intrinsic evaluation to assess the quality and see if the merged dataset holds the similar quality as XL-Sum dataset. The intrinsic evaluation assesses certain, measurable characteristics of summaries. There are many automatic metrics to quantify important features of abstractive summaries (e.g., abstractivity, novel n-grams, compression, redundancy, BERTScore, BLEU).

4.5.1 Abstractivity:

The amount of the content in the summary that is newly generated (distinct from copied) content from the source document is called abstractivity. By being more abstractivity, the dataset indicates that a model can learn to generate new content rather than just obtain text from it. Abstractivity is measured by finding fragments that are greedily matched between the input article and the summary [18] [11]. Abstractivity is measured by

$$ABS_p(D_i, S_i) = 1 - \frac{\sum_{f \in \mathcal{F}(D_i, S_i)} |f|^p}{|S_i|^p} \tag{4.1}$$

4.5.2 Compression:

Compression refers to the amount of how much information in the document is summarized in a short length. A higher compression score is required. The higher the score, the shorter the summary and yet it includes the same information. Compression is calculated as the ratio of the length of the text and of the summary[18]. Compression is measured by

$$\mathrm{CMP}(A, S) = 1 - \frac{|S|}{|A|} \tag{4.2}$$

4.5.3 Novel n-grams:

The extent to which new words or sequences of words (n-grams) appear in the summary compared to the text is measured by novel n-grams: uni-gram measures novel words, bi-gram, tri-gram for novel two and three word sequences respectively. On abstractive summarization datasets, higher percentages of novel n-grams are better [12]. Novel n-grams are measured by

$$\text{Novel n-grams} = \frac{\text{Number of Novel n-grams}}{\text{Total Number of n-grams in Summary}} \tag{4.3}$$

4.5.4 Redundancy:

The redundancy measures the amount of repetition of information in the summary, uni-gram redundancy is to repeat the common single words, bi-gram redundancy is to repeat the pairs of words. Lower redundancy scores are better for abstractive summarization datasets because they mean the summarization contains less repetition, which improves readability and conciseness [23]. Redundancy is measured by

$$\text{RED}(S) = \frac{\sum_{i=1}^{m}(f_i - 1)}{\sum_{i=1}^{m} f_i} \tag{4.4}$$

Metric	**XL-Sum Dataset**	**Daily Star Dataset**	**Merged Dataset**
Abstractivity (%)	72.76	70.81	71.60
Compression (%)	94.74	88.02	92.78
Novel Uni-grams (%)	38.81	48.84	43.12
Novel Bi-grams (%)	81.10	80.91	81.30
Novel Tri-grams (%)	92.10	91.47	92.40
Redundancy (Uni-grams) (%)	2.93	3.35	3.55
Redundancy (Bi-grams) (%)	0.25	0.35	0.34

Table 4.2: Comparison Of Different Metrics Across Different Datasets

From the table we can see the metrics for the Merged dataset are generally close to those of the XL-Sum dataset across most of the evaluated parameters. The Merged dataset has an abstractivity value of 71.60 %, slightly less than 72.76 percent for XL-Sum. In the Merged dataset, the compression score is 92.78% compared to 94.74% for the XL-Sum compression score is close to the same level of conciseness. Regarding novel n-grams, in the Merged dataset uni-gram shows a value of 43.12%, which is somewhat higher compared to 38.81% for XL-Sum, while for bi-gram the values are quite close to 81.30% for Merged dataset and 81.10% for XL-Sum. The tri-gram also has minor differences, with the Merged dataset having 92.40% compared to 92.10% for XL-Sum. The redundancy values for uni-gram and bi-gram metrics also indicate a close match. The uni-gram redundancy for the Merged dataset is 3.55%, while for XL-Sum it is 2.93%. Similarly, the bi-gram redundancy shows a minimal difference, with 0.34% for Merged compared to 0.25% for XL-Sum. These small differences across the metrics suggest that the Merged dataset can hold some similar qualities as the XL-Sum dataset.

4.5.5 BERTScore:

BERTScore is a natural language generation evaluation metric, including summarization. BERTScore is very well applicable to the summary generation task of paraphrasing,

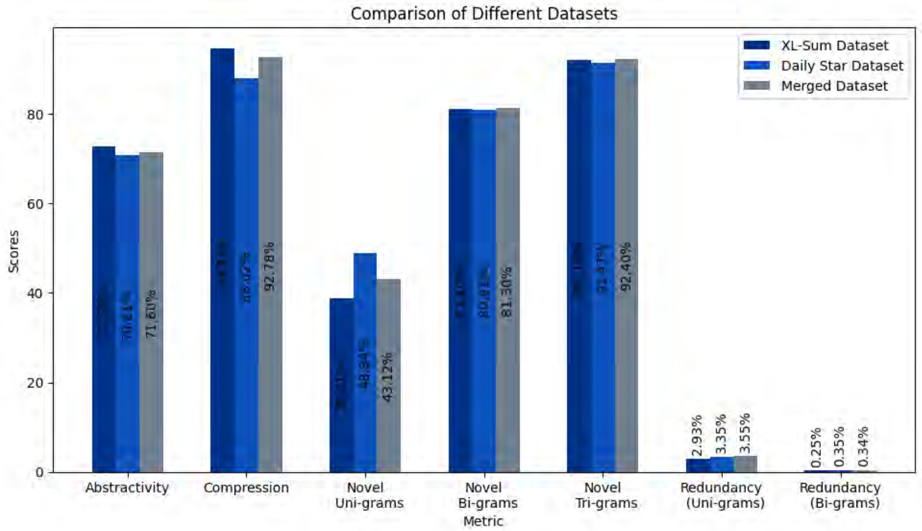


Figure 4.15: Comparative Analysis of Abstractivity, Compression, Novel n-grams, Redundancy Across Different Dataset

rephrasing. Considering semantic meaning which is essential to abstractive summarization, BERTScore can better evaluate the summaries. It uses precision, recall and the F1 score. The precision entails the amount of the summary relevant to the text. The recall measures how much of the article content is contained in the summary. And the harmonic mean of precision and recall is F1 score. This score tells us how well the summary matches the original text in terms of both what it includes and how relevant that information is [27] [17].

Metric	XL-Sum Dataset	Daily Star Dataset	Merged Dataset
Precision(%)	73.63	74.82	74.05
Recall(%)	62.00	63.01	62.59
F1 Score(%)	67.28	68.37	67.80

Table 4.3: BERTScore Across Different Datasets

From the table we can see the Merged dataset reaches 74.05% precision and 73.63% for XL-Sum which means the quality of the generated summary in terms of relevance is similar. The Coverage of the content of the merged dataset is almost the same as XL-Sum, because recall of the merged dataset is 62.59%, and 62.00% for XL-Sum. The F1 Score for Merged dataset is 67.80% and that of XLSum dataset is 67.28% which is balanced over precision and recall across both datasets. These small differences imply similar inherent quality, relevance, coverage properties for the merged dataset compared to XL-Sum dataset. The fact of this close alignment means that the quality attributes of the XL-Sum dataset are preserved in the merged dataset.

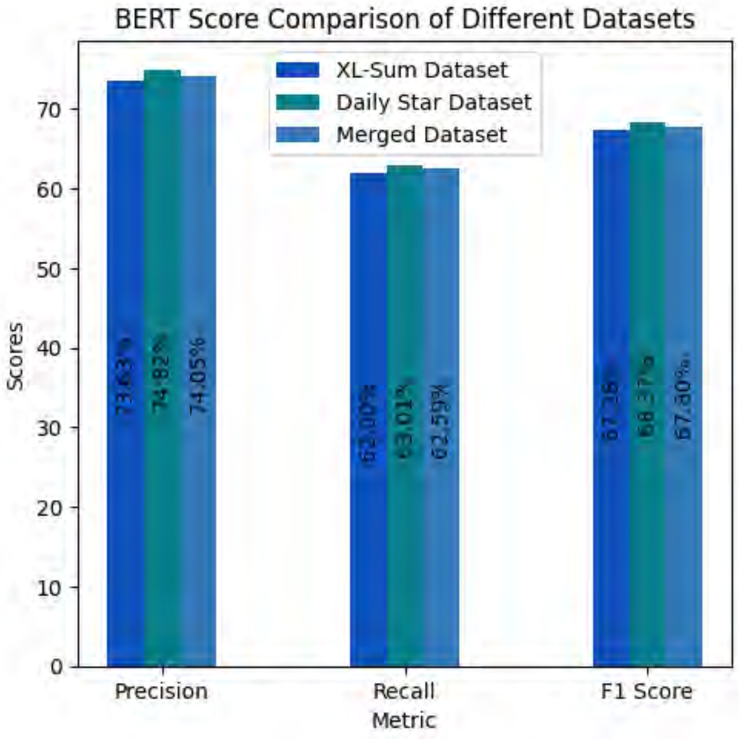


Figure 4.16: Comparative Analysis of BERTScore Across Different Dataset

4.5.6 BLEU

Until now, we have evaluated different types of metrics to understand how much abstractive the dataset is by measuring semantic richness, and novelty of content, as well as how well the generated summaries reflect the meaning of the original text instead of just copying. These also help to understand the quality of the summaries in terms of coverage. Now we will evaluate the BLEU metric which is commonly used for summarization. For this case, low BLEU values represent abstractive summaries as it only sees exact words.

BLEU (Bilingual Evaluation Understudy) is an evaluation metric to see how well the text generated is similar to the reference text by using n-gram overlap. The BLEU measures the number of n-gram (e.g. uni-gram, bi-gram, tri-gram) from the summary that overlaps the n-gram of the reference text.The uni-gram precision is used to observe how much the each word of the summary document matches with the reference document. Bi-gram is for two-word sequences between the generated summaries and the reference text. Tri-gram precision is for three-word sequences [1].

Metric	XL-Sum Dataset	Daily Star Dataset	Merged Dataset
Uni-gram(%)	2.07	2.18	2.44
Bi-gram(%)	1.06	1.21	1.26
Tri-gram(%)	0.65	0.76	0.77

Table 4.4: BLEU Score Across Different Datasets

The table shows that the n-gram values for XL-Sum and the merged dataset are extremely low.The merged data's n-gram values are close to the values of the XL-Sum dataset. Low BLEU scores among the various n-gram levels suggest that the merged datasets are highly abstractive, similar to the XL-Sum dataset. Consequently the summaries in the merged dataset do not closely represent the reference texts in terms of exact wording and word sequence, which are typical traits of the abstractive summarization.

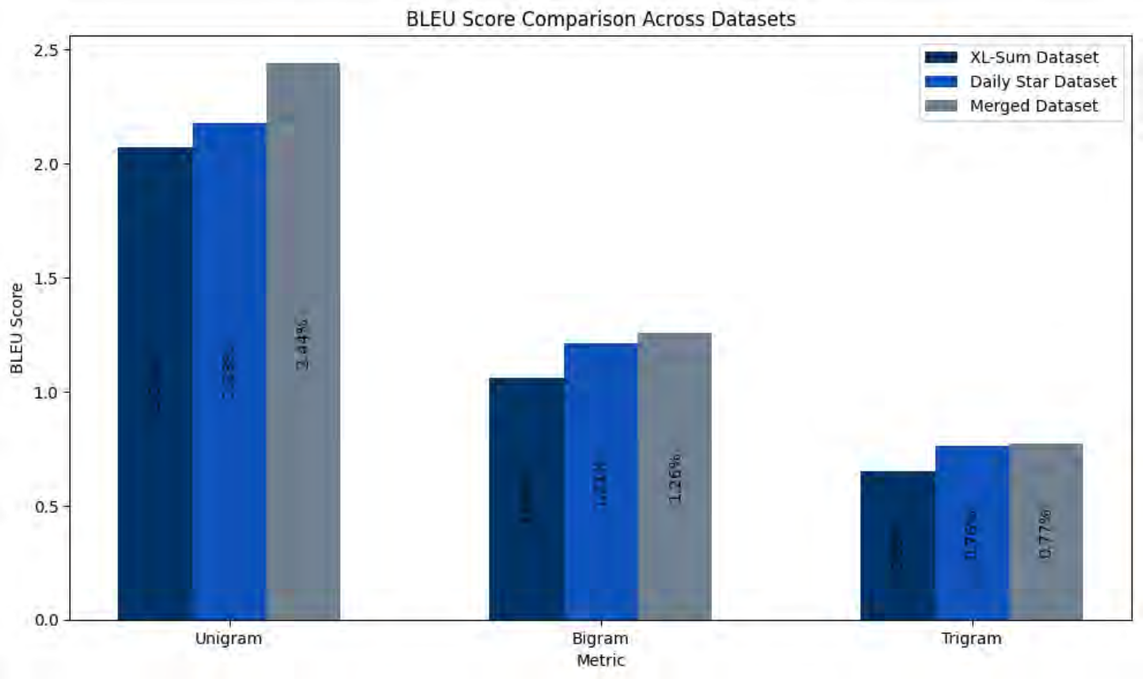


Figure 4.17: Comparative Analysis of BLEU Across Different Dataset

Chapter 5

Model Description

5.1 T5

The model T5 (Text-To-Text Transfer Transformer) is a type of deep learning model which is basically focused on Natural Language Processing tasks. It is a Encoder-Decoder model and it converts all NLP problems into text-to-text format. It means the input and output of this model is both in text format. This model is pre-trained on the C4 corpus which is a very large dataset containing clean text extracted data from web pages. Pre-training on this large corpus helps this model to learn the language patterns and many other things that play an important role while working on text.

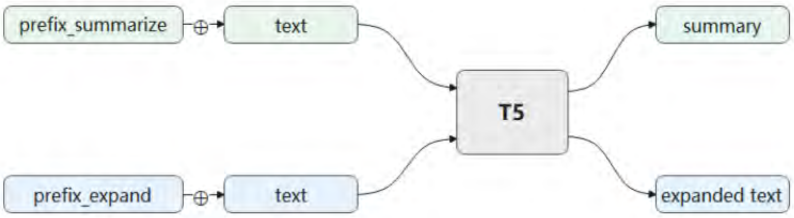


Figure 5.1: T5 model with different prefixes

This figure [5.1] [28] shows how this model works on different prefixes. It receives the prefix which is basically an instruction of what type of task that the model needs to do. Along with the input and prefix, this model next does the task which has been instructed for example: summary, expansion of input text etc.

This figure [5.2] [20] is a more detailed view of what types of tasks the T5 model does. The first type of task in the figure shown is translation. Here the model translated the following input according to the instruction. The second type of task is, linguistic acceptability and understanding of any input. In this part, it is checked that an input is grammatically correct or not or the following sentence is linguistically accepted or not. The third one is an example of semantic textual similarity between two sentences. Here

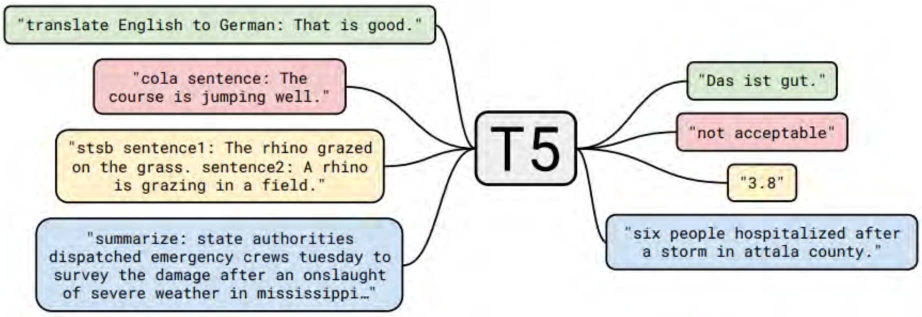

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Figure 5.2: Framework of the T5 model

the output is a number where the range of the output is 0 to 5. This model compares between the input sentences and it produces output in between that range. The last one is a summarization task. Where the input is a quite large document and the produced output is a summary of the input. This model does this kind of textual task while the format of input is, giving the actual input or sentences on which the model works along with the prefix, which is mentioning what type of task the model has to be done.

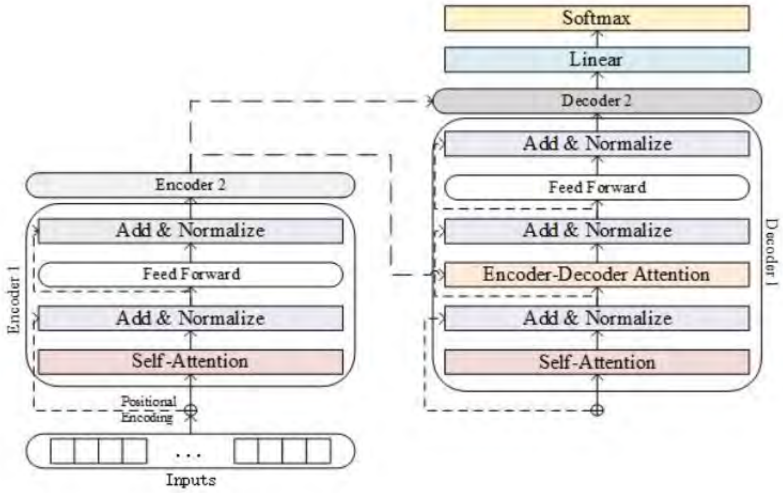


Figure 5.3: Architecture of the T5 model

This figure[5.3][28] is presenting the architecture of the T5 model. From the Inputs part, the model gets the input on which it will be working on. In this part, tokenization is completed meaning the whole input text is splitted into smaller pieces so that the model is able to do the necessary processes with these tokens. As transformers do not contain any track of token order, each token gets its position in the sequence from the positional encoding part. This model is a combination of a bunch of encoder-decoders. After receiving position from the positional encoding, the encoder part starts to process the input text and also tries to understand the context of every word. Each block contains layers of self-attention and feed forward neural network. The importance of every word comparing each other is calculated in the self-attention process. And feed forward is a neural network which processes the output of the attention layer. Each sub layer which consists of self-attention and feed forward neural network is connected with a residual connection which is add and normalize. It is followed by layer normalization and controls the stability of the learning process. The Decoding part is also made up with a bunch of decoders where each layer contains self-attention, encoder-decoder attention and feed-forward neural networks. Similar to the encoding part of self-attention, in this part of decoding it limits what each place in the output sequence can do to what comes before it. The encoder - decoder attention part works for the decoder to focus on relevant parts of input sentences. And the feed forward neural network processes the output in that similar way of the attention layer. The linear layer is basically a representation of the vocabulary of the model which converts the output of the decoder to a higher - dimensional space. And lastly, the softmax is a probabilistic function which is applied to the linear layer output for every token of the vocabulary to determine the next token.

5.2 BART

In Natural Language Processing (NLP) one of the notable models is BART (Bidirectional and Auto-Regressive Transformers). BART is an encoder decoder network which shares similarity with BERT and GPT models. Over small supervised datasets, the BART models can be fine tuned to create domain specific tasks. BART has multiple encoder and decoder layers.

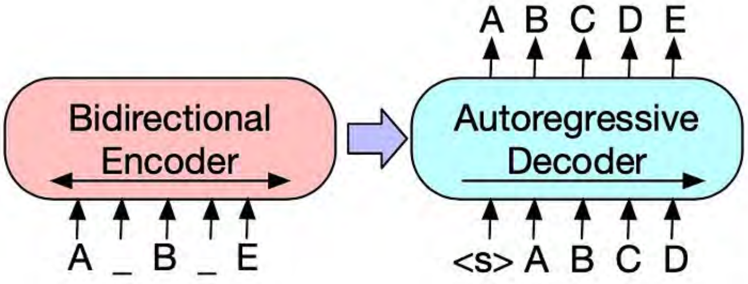


Figure 5.4: Bidirectional Encoder And Autoregressive Decoder [15]

It is an autoencoder, a form of denoising autoencoder. This is a neuron network which learns useful features by encoding degraded input sentences and decoding it back into its original form. The model is used for text summarization, question answering , and language translation. The encoder is similar to BERT and is bidirectional (allowing it to understand context in both directions) and the decoder similar to GPT generates the output in an autoregressive manner (token by token).

Each encoder layer has many components. Before entering input embedding the tokenization is completed. Then in input embedding the input tokens are converted into numerical vectors. Furthermore in positional encoding, the positional information is added to these embeddings because the transformer model doesn't inherently understand the order of words. The multi head attention mechanism after that allows the encoder to attend to

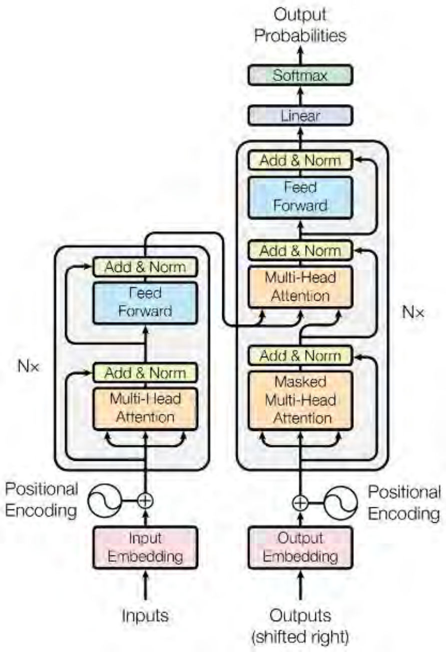


Figure 5.5: BART single encoder-decoder network architecture [29]

different parts of the sentence and analyze relations of words (in a bidirectional manner). So that the encoder will be able to capture both preceding and succeeding words, and that's really what makes it effective in understanding the entire context. After the multi head attention mechanism, the encoder adds the original input to the output of the attention layer through skips connections. Then the layer normalization is applied next to stabilize the network and maintain consistency of values across tokens, making its way to all tokens uniformly processed. The process of outputs being further processed to the feed forward neural networks provides further representations to each token. And then the output is just a sequence of contextual embeddings, for each input token ready to feed to the decoder. BART has a decoder which goes autoregressive where it goes token by token, given the previous tokens it is trying to predict the next one. First output Embedding takes the shifted right output sequence. Similar to the encoder, positional encodings are added to maintain the correct word order in the positional encoding layer. In the decoder part, as the current output tokens, the Masked Multi Head Attention layer processes them and ensures that the model does not see any future tokens. It helps predict one token at a time, by only attending to the already generated tokens. Furthermore, the decoder is connected to the encoder output by the Cross-Attention layer. This layer takes the encoder's embeddings. By linking up the created output with the input sequence, it gathers relevant information. Like the encoder, the output from both the masked self attention as well as the cross attention layers is added back through skip connections and further layer normalized in add and norm layer. Then the linear layer processes the output that is produced by decoder. And the layer converts the output in logits. Then the softmax layer transforms them in probabilities and makes sure every probability is in a range. Based on the highest probability the next word is chosen [15].

5.3 Llama 3.1

Llama (Large Language Model Meta AI) 3.1 is a transformer-based language model which is developed by Meta AI with a focus on various natural language processing (NLP) tasks, more specifically where generation is the main goal such as text summarization, translation, question answering etc. Llama 3.1 is very much effective as it was trained over 15 trillion tokens, at the same time continuous optimization for scalability and efficiency. Utilizing over 16,000 H100 GPUs ensures a very high quality performance and resource management of this model.

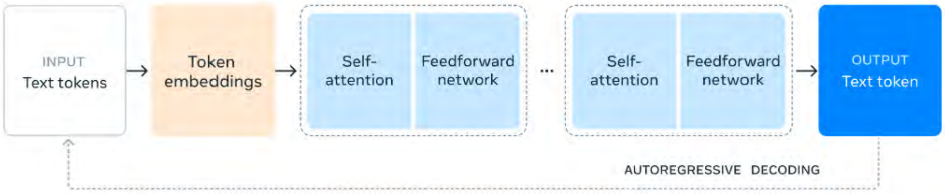


Figure 5.6: Architecture of Llama 3.1 [30]

Llama 3.1 is a decoder-only transformer model whose architecture is particularly very effective for the autoregressive tasks. In autoregressive tasks, models predict the next word in a sequence based on the whole context. In text generation tasks, this model performs in this way. The first step of Llama 3.1 is the transformation of input which is basically text. This step is token embedding, where each input text is represented as a high-dimensional vector with capturing the semantic and syntactic properties of the text. To generate meaningful text with actual context, capturing the semantic and syntactic properties are very important. But this doesn't include remembering the positions of tokens. Although it's very important to contain the actual meaning of the input text. That is the reason why positional encodings are added to the token embeddings, which ensure to keep track of the position of the generated tokens in a sequence. The next step is Self-Attention Mechanism which is the core of the Llama 3.1 model for its multi-head mechanism. This mechanism ensures Llama 3.1 to focus on different parts at the same time of the input text. Because of this mechanism Llama can capture both long and short term dependencies between words in a sequence. Capturing these long and short term dependencies ensures to generate more relevant and coherent text. After processing the information in self-attention, these are passed through feed-forward networks. By passing through this layer, the model learned the representations while providing additional abstraction. This process enhances the ability to generate from the input text. For deeper understanding of input text and to ensure more coherent and contextual output there can be more occurrences of this layer. To make sure the model is learning efficiently normalization is applied in various points of the model. To pass the information efficiently between layers and prevent the degradation of learned features residual connections are employed. The last step is the output layer which is basically the generation of desired text. This layer produces probabilities for every next token in autoregressive fashion based on the processed input and generates one word at a time.

5.4 Bloom 576M

Bloom (BigScience Large Open-science Open-access Multilingual Language Model) is a decoder-only transformer model which is designed for natural language processing (NLP) tasks such as text generation, translation, question answering, sentiment analysis etc. With a capability of zero-shot and few-shot learning, this model has the ability to translate across 46 natural languages and 13 programming languages.

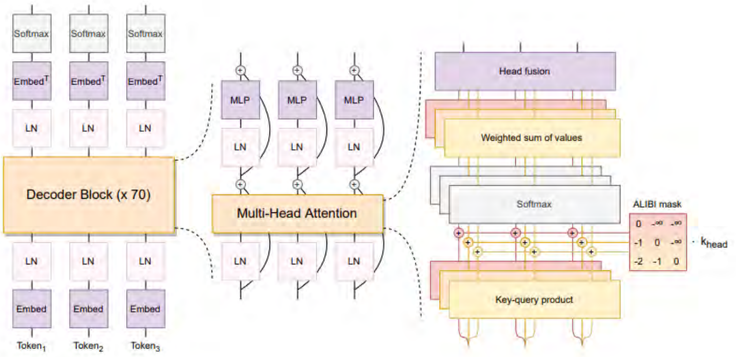


Figure 5.7: Architecture of Bloom [26]

This diagram represents the BLOOM architecture. This model follows a transformer architecture with several important designs which ensures optimizing both performance and training stability. Bloom is built with the use of 70 stacked decoder blocks, where each of them processes the input tokens by passing through multiple transformer layers. In each layer there are self-attention and feed-forward sub-layers. Before going to the next layer the generated tokens go through layer normalization. This model uses a multi-head self-attention mechanism which basically allows one to focus on different parts of the input sequence simultaneously. Because of this it can capture dependencies in a long sequence more efficiently. In this mechanism, a key-query product is calculated which is followed by a softmax layer and applies an ALiBi mask. The main purpose of the mask is to adjust attention weights which are calculated based on the distance between tokens and it ensures the performance on longer sequences. The outputs from the attention mechanism are passed through feed-forward networks (MLP), which ensures further abstraction and transformation of the input data for generating more relevant and coherent text, also training stability and smooth propagation of gradients. To adjust attention wights BLOOM uses AliBi (Attention with Linear Biases) positional embeddings, which ensures better performance for longer sequences. In the Head Fusion step, all outputs from all the heads which are the attention scores combined into a single vector. After that a weighted sum of values is calculated by using the attention scores as weights and compute the sum of values. Here the most relevant parts of the inputs are combined which generate more informed representation for next processing. After that, these values are passed through the subsequent layers of the model to generate the final output. For example in text generation, the output is predicting the next token in the sequence.

Chapter 6

Result Analysis

6.1 ROUGE Score

In our research we used ROUGE score to evaluate the fine-tuned models. We used ROUGE-1 , ROUGE-2 and ROUGE-L metrices. Those ROUGE metrices and ROUGE scores are based on the n-grams concepts. ROUGE-1 measures the overlap of Uni-grams between the model generated summaries and the human generated reference summaries of the test dataset. ROUGE-2 measures the overlap of Bi-grams and ROUGE-L measures the Longest Common Sub-sequences. ROUGE-L doesn't necessarily need consecutive matches, it can work with in-sequence matches.

Model	Batch Size	Epoch	ROUGE-1	ROUGE-2	ROUGE-L
mT5 (Collected Dataset)	32	1	20.92	7.60	18.49
mT5 (Our Dataset)	32	1	21.01	7.58	18.36
BanglaT5	32	10	16.19	5.24	14.49
mBART	32	10	14.49	4.03	11.98
Llama 3.1	-	-	13.10	3.71	11.50
Bloom-576M	8	5	7.81	1.62	5.55

Table 6.1: Model Performance(ROUGE Score)

We have fine-tuned mT5, BanglaT5, mBART and Bloom-576M models. Firstly, we have fine-tuned the mT5 model with the collected dataset and our dataset. After training the model with these two datasets we tasted the model with the same testing dataset to evaluate them. The ROUGE-1 , ROUGE-2 and ROUGE-L generated by mT5(collected dataset) consecutively are 20.92, 7.60 and 18.49. The ROUGE-1 , ROUGE-2 and ROUGE-L generated by mT5(Our dataset) consecutively are 21.01,7.58 and 18.36. Here, the mT5 model fine-tuned by our dataset beats the mt5 model fine-tuned by the collected dataset on the ROUGE-1 score. The ROUGE-2 score is almost similar. But on the ROUGE-L score mt5 fine-tuned by the collected dataset is higher than the mT5 model fine-tuned by our dataset.

Then we fine-tuned more models to evaluate which model can perform better in Bangla summarization. The fine-tuned BanglaT5 model has generated ROUGE-1 , ROUGE-2 and ROUGE-L scores consecutively of 16.19, 5.24 and 14.49. Which is lower than the mT5 model fine-tuned by the both datasets. But higher than other models we fine-tuned. The fine-tuned mBART model has generated ROUGE-1 , ROUGE-2 and ROUGE-L scores consecutively of 14.49, 4.03 and 11.98. Another model we fine-tuned is Bloom-576M . This is one of the two LLM models we tried for generating Bangla summary. The ROUGE-1 , ROUGE-2 and ROUGE-L scores generated by this model consecutively are 7.81 , 1.62 and 5.55. The scores generated by the Bloom-576M model are so low because this model didn't generate a summary of the given text. It has generated a passage based on the given text.

Lastly we used another LLM model named Llama 3.1 to generate Bangla Summary. We didn't train this model with our dataset because the model is created for doing multiple tasks and training it with another dataset is beyond our computational capacity. We used prompt engineering(langchain) to make the model generate summaries. We used 4 bit quantization because using the whole model is beyond our computational capacity. The ROUGE-1 , ROUGE-2 and ROUGE-L scores generated by this model consecutively are 13.10, 3.71 and 11.50. These results can be improved if we can increase the computational capacity.

However, ROUGE has some limitations when it comes to abstractive summarization and semantic accuracy. In the case of abstractive summary, summaries are generated using completely different words or sentence structures that have the same meaning. ROUGE tends to fail in measuring the evaluation of those summaries since it relies on word overlap. Bangla is a rich language which has many synonyms of one word. Bangla is also a morphological language with various sentence structures and word forms. Because of this the calculation of word overlap faces many mismatches which causes lower ROUGE value [7].

6.2 Other scores

In order to find out the quality of the summaries generated by the models. We evaluated the summaries in Abstractivity, Compression, Novel Uni-grams, Novel Bi-grams, Novel Tri-grams, Redundancy(Uni-grams) and Redundancy(Bi-grams) metrices.

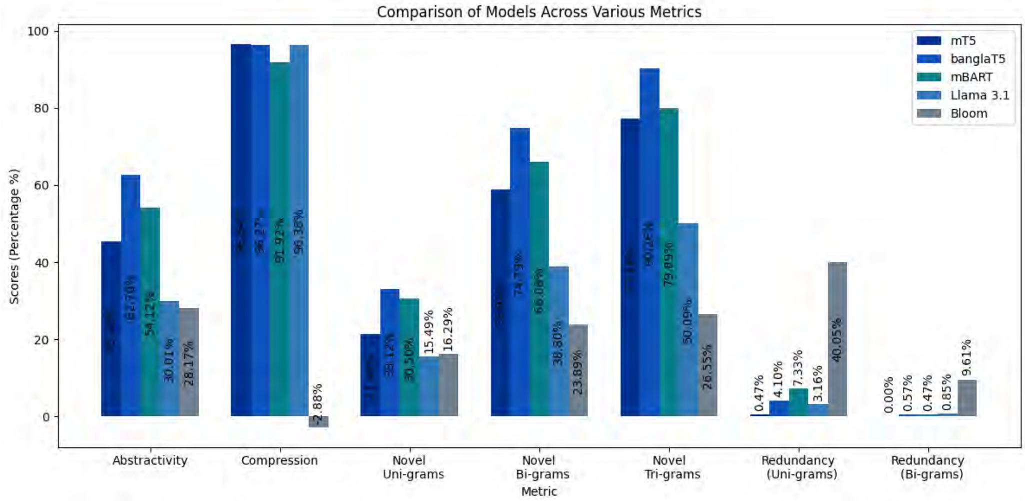


Figure 6.1: Comparison of Models Across Various Metrices

6.2.1 Abstractivity

The Abstractivity metric score describes how much abstractive the generated summaries are. Since we are working on abstractive summarization this score is very important. In terms of Abstractivity metric summaries generated by the BanglaT5 model has the highest score. It has a score of 62.70% . Then comes the summaries generated by the mBART model, it has a score of 54.12%. Both of the models beat the score of the mT5 model, which is 45.32%. The Llama 3.1 model has an Abstractivity score of 30.01% and the Bloom-576M has a score of 28.17%. From those scores we can say that the summaries generated by BanglaT5 and mBART models are more abstractive than the mT5 model. But summaries generated by Llama 3.1 and Bloom-576M models are less abstractive than the mT5 model.

6.2.2 Compression

The compression metric score shows how much shorter the generated summaries are compared to the reference text. The higher the compression score, the shorter the summary and yet it includes the same information. Here the summaries generated by mT5, BnaglaT5 and Llama 3.1 have almost similar scores consecutively 96.64%, 96.27% and 96.38%. The mBART model has a slightly lower score than these three models , which is 91.92%. Where the Bloom-576M model has a negative score of 2.88% , which means the Bloom-576M model generates expanded passages rather than summaries.

6.2.3 Novel Uni-grams, Novel Bi-grams and Novel Tri-grams

Novel Uni-grams, Bi-grams and Tri-grams in a summary describes how many unique words, two worded phrases and sequence of three words are present in the summary. The higher score in these metrices are better for abstractive summaries. The BanglaT5 model has the highest score in all three of the metrices. The novel Uni-grams, Bi-grams and Tri-grams scores of the BanglaT5 model are 33.12%, 74.79% and 90.26%. After BanglaT5 , the mBART model has best scores , which are 30.50% in novel Uni-grams, 66.08% in novel Bi-grams and 79.89% in novel Tri-grams metrices. Both BanglaT5 and mBART have beaten the mT5 model in Novel Uni-grams, Bi-grams and Tri-grams scores. The mT5 model has novel Uni-grams score of 21.49%, novel Bi-grams score of 58.94% and novel Tri-grams score of 77.14%. Novel Uni-grams, Bi-grams and Tri-grams scores of the Llama 3.1 model consecutively are 15.49%, 38.80% and 50.09%. Lastly, the Bloom-576M model has lowest scores in these metrices. Novel Uni-grams, Bi-grams and Tri-grams scores of the Bloom-576M model consecutively are 16.29%, 23.89% and 26.55%.

6.2.4 Redundancy(Uni-grams and Bi-grams)

The redundancy measures the amount of repetition of information in the summary, Uni-grams redundancy is to repeat the common single words, Bi-grams redundancy is to repeat the pairs of words. Lower redundancy score is better for abstractive summary.

The mT5 model has a lower redundancy score in Uni-grams and Bi-grams than other models. The Uni-grams redundancy score is 0.47% and Bi-grams redundancy is 0%. BanglaT5 is slightly higher with the Uni-grams redundancy score of 4.10% and Bi-grams redundancy of 0.57%. The mBART model also has very low redundancy values. In case of Uni-grams redundancy The mBART model has a score of 7.33% and in Bi-grams redundancy it has a score of 0.47%. The Llama 3.1 model has beaten BanglaT5 and mBART models in case of Uni-grams redundancy with a score of 3.16%. The Bi-grams redundancy score is also very close to other models, which is 0.85%. But the Bloom-576M has a Uni-grams redundancy score of 40.05% , which is very high. It also has a high Bi-grams redundancy of 9.61%.

From the scores of these metrices we can see that the summaries generated by BanglaT5 model have beaten all other models in abstractivity, Novel Uni-grams, Bi-grams and Tri-grams. It also has lower values in Uni-grams and Bi-grams redundancy. The compression value is also higher and close to other models. The mBART model also beat the base mT5 model in case of abstractivity, Novel Uni-grams, Bi-grams and Tri-grams.It also has lower values in Uni-grams and Bi-grams redundancy. The compression value is also higher and close to other models. The Llama 3.1 model has a very good score in compression higher than mBART and BanglaT5 , but it has fallen behind in other metrices. From the results we can say that BanglaT5 and mBART is better than the base model mT5 in case of abstractivity, Novel Uni-grams, Bi-grams and Tri-grams. Those models have very good scores in compression, Uni-grams and Bi-grams redundancy, but slightly lower in compression metrices and slightly higher in Uni-grams and Bi-grams redundancy metrices than mT5 model. Lastly, the abstractivity, compression, Novel Uni-grams, Novel Bi-grams ,Novel Tri-grams and redundancy score of the Bloom-576M model shows that it has generated extended passages rather than summaries.

6.3 BERTScore

BERTScore can better evaluate the summaries considering semantic meaning which is essential to abstractive summarization. The BERTScore precision shows the amount of the summary relevant to the text. The BERTScore recall is how much of the article content is contained in the summary. And the harmonic mean of precision and recall is F1 score. This BERTscore F1 tells us how well summary matches the original text in terms of both what it includes and how relevant that information is.

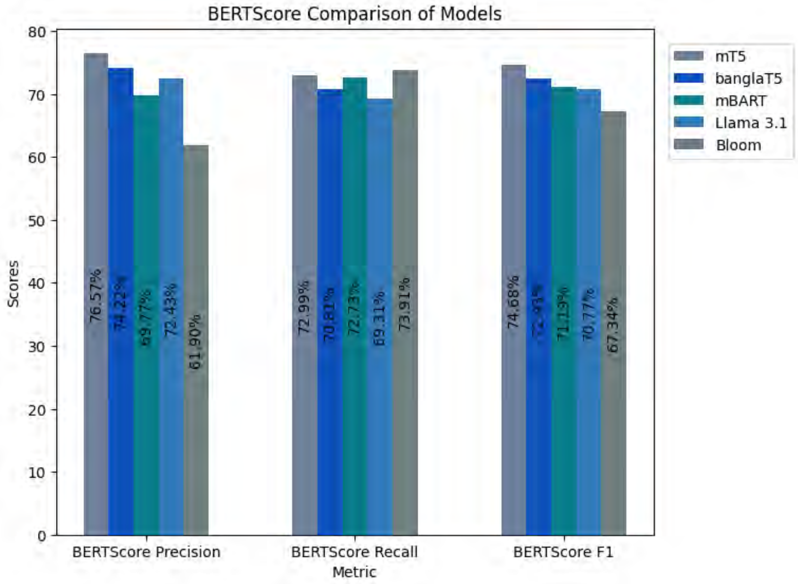


Figure 6.2: BertSCORE Comparison of Models

From the barcharts we can clearly see that the mT5, BanglaT5, mBART and Llama 3.1 models have good scores in BERTScore precision which means the summary generated by those models is very relevant to the text. The mT5 model has precision score of 76.57% , BanglaT5 model has precision score of 74.22%, mBART model has precision score of 69.77% and Llama 3.1 has precision score of 72.43%. Only Bloom-576M has a lower score of 61.90%.

In case of BERTScore recall all the models have scores above 70% except Llama 3.1, it has a score of 69.31% which is also close to 70%. So all the models cover a good amount of article's information in the summary. The mT5 model has recall score of 72.99% , BanglaT5 model has recall score of 70.81%, mBART model has recall score of 72.73% and Bloom-576M has a recall score of 73.91%.

Lastly, in the case of BERTScore F1 score all the models have a good F1 score ,which means how well summary matches the original text. Here mT5 beats other models with the score of 74.68%, but others models are close to the score. Summaries generated by BanglaT5 have a score of 72.43% F1 score. The mBART model has an F1 score of 71.19%. Llama 3.1 has 70.77 and Bloom-576M has a score of 67.34%.

From all the scores we can say that all the models have almost similar semantic meaning. Since the BERTScore precision, recall and F1 score of all the models are close to each other.

6.4 Example of summaries generated by the models

Text	সিরাম ইনস্টিটিউট থেকে বাংলাদেশে প্রতিমাসে ৫০ লাখ ডোজ করে টিকা আসার কথা ছিল স্বাস্থ্য অধিদপ্তর একটি বিজ্ঞপ্তিতে জানিয়েছে, ২৬শে এপ্রিল সোমবার থেকে পরবর্তী নির্দেশ না দেয়া পর্যন্ত কোভিড-১৯ টিকাদান কার্যক্রমের প্রথম ডোজ টিকা প্রদান সাময়িকভাবে বন্ধ থাকবে । স্বাস্থ্য অধিদপ্তরের তথ্য অনুযায়ী, বাংলাদেশে ২৪শে এপ্রিল পর্যন্ত ৫৭ লাখ ৯৮ হাজার ৮৮০ জন প্রথম ডোজের টিকা নিয়েছেন। আর দ্বিতীয় ডোজের টিকা পেয়েছেন ২১ লাখ ৫৫ হাজার ২৯৬ জন। টিকা গ্রহণের জন্য এ পর্যন্ত নিবন্ধন করেছেন ৭২ লাখ ৬ হাজার ৫৬৫ জন। বাংলাদেশ এমন সময় এই সিদ্ধান্ত নিয়েছে, যখন ভারত টিকা রপ্তানির ওপর নিষেধাজ্ঞা আরোপ করায় দেশটি এক গভীর সংকটে পড়েছে। বেসরকারি সংস্থা বেক্সিমকোর মাধ্যমে ভারতের সিরাম ইন্সটিটিউটে তৈরি করা অক্সফোর্ড-অ্যাস্ট্রজেনেকার কোভিশিল্ড তিন কোটি ডোজ টিকা সংগ্রহের উদ্যোগ নিয়েছিল সরকার। কিন্তু প্রতিষ্ঠানটি থেকে এখন পর্যন্ত মাত্র ৭০ লাখ ডোজ টিকা পাওয়া গেছে। আরও ৩২ লাখ টিকা সরকার পেয়েছে উপহার হিসাবে। সেই হিসাবে বাংলাদেশের মোট এক কোটি ২ লাখ ডোজ টিকা পেয়েছে, যার একটি বড় অংশই দেয়া হয়ে গেছে। এমন প্রেক্ষাপটে প্রথম ডোজ টিকা দেয়া সাময়িকভাবে বন্ধ রাখার সিদ্ধান্ত নিয়েছে সরকার।
Original Summary	বাংলাদেশে আগামীকাল সোমবার থেকে করোনাভাইরাসের টিকার প্রথম ডোজ দেয়া বন্ধ করে দিচ্ছে কর্তৃপক্ষ।
Summary Generated by mT5	বাংলাদেশে কোভিড-১৯ টিকার প্রথম ডোজ দেয়া কার্যক্রম বন্ধ করার সিদ্ধান্ত নিয়েছে সরকার।
Summary Generated by mBART	বাংলাদেশে করোনাভাইরাসের টিকা দেয়া সাময়িকভাবে বন্ধ রাখার সিদ্ধান্ত নিয়েছে সরকার। কিন্তু সরকার বলছে, গত ২৪শে এপ্রিল থেকে টিকার সংখ্যা এক কোটি ছাড়িয়ে যাবার জন্য টিকা নেয়া সব মানুষের জন্যে যথেষ্ট টিকাই দেয়া যাবে।
Summary Generated by BanglaT5	বাংলাদেশে করোনাভাইরাসের টিকাদান কর্মসূচি বন্ধ রাখার সিদ্ধান্ত নিয়েছে সরকার।
Summary Generated by Llama 3.1	সিরাম ইনস্টিটিউট থেকে বাংলাদেশে প্রতিমাসে ৫০ লাখ ডোজ করে টিকা আসার কথা ছিল স্বাস্থ্য অধিদপ্তর একটি বিজ্ঞপ্তিতে জানিয়েছে, কিন্তু প্রথম ডোজ টিকা দেয়া সাময়িকভাবে বন্ধ রাখার সিদ্ধান্ত নিয়েছে সরকার।

Summary Generated by Bloom-576M	সিরাম ইনস্টিটিউট থেকে বাংলাদেশে প্রতিমাসে ৫০ লাখ ডোজ করে টিকা আসার কথা ছিল স্বাস্থ্য অধিদপ্তর একটি বিজ্ঞপ্তিতে জানিয়েছে, ২৬শে এপ্রিল সোমবার থেকে পরবর্তী নির্দেশ না দেয়া পর্যন্ত কোভিড-১৯ টিকাদান কার্যক্রমের প্রথম ডোজ টিকা প্রদান সাময়িকভাবে বন্ধ থাকবে। স্বাস্থ্য অধিদপ্তরের তথ্য অনুযায়ী, বাংলাদেশে ২৪শে এপ্রিল পর্যন্ত ৫৭ লাখ ৯৮ হাজার ৮৮০ জন প্রথম ডোজের টিকা নিয়েছেন। আর দ্বিতীয় ডোজের টিকা পেয়েছেন ২১ লাখ ৫৫ হাজার ২৯৬ জন। টিকা গ্রহণের জন্য এ পর্যন্ত নিবন্ধন করেছেন ৭২ লাখ ৬ হাজার ৫৬৫ জন। বাংলাদেশ এমন সময় এই সিদ্ধান্ত নিয়েছে, যখন ভারত টিকা রপ্তানির ওপর নিষেধাজ্ঞা আরোপ করায় দেশটি এক গভীর সংকটে পড়েছে। বেসরকারি সংস্থা বেক্সিমকোর মাধ্যমে ভারতের সেরাম ইন্সটিটিউটে তৈরি করা অক্সফোর্ড-অ্যাস্ট্রজেনেকার কোভিশিল্ড তিন কোটি ডোজ টিকা সংগ্রহের উদ্যোগ নিয়েছিল সরকার। কিন্তু প্রতিষ্ঠানটি থেকে এখন পর্যন্ত মাত্র ৭০ লাখ ডোজ টিকা পাওয়া গেছে। আরও ৩২ লাখ টিকা সরকার পেয়েছে উপহার হিসাবে। সেই হিসাবে বাংলাদেশের মোট এক কোটি ২ লাখ ডোজ টিকা পেয়েছে, যার একটি বড় অংশই দেয়া হয়ে গেছে। এমন প্রেক্ষাপটে প্রথম ডোজ টিকা দেয়া সাময়িকভাবে বন্ধ রাখার সিদ্ধান্ত নিয়েছে সরকার।

We can see from the example that summaries generated by mT5, BanglaT5, mBart and Llama 3.1 models are very concise and have the main concept of the text. But the Bloom-576M model generates a whole passage rather than a summary of the text.

Chapter 7

Error Analysis

We manually analyzed the generated summaries to see their overall quality. This will help to know the area where the summarizing procedure needs to be improved.

- **T**he length of some generated summaries were longer or shorter than the reference summary length.

  Example:

Original Summary	গত বেশ কিছুদিন বাংলাদেশের রাজধানী ঢাকায় বাসিন্দাদের আলোচনার প্রধান বিষয় দু:সহ গরম।
Summary Generated by mT5	বাংলাদেশে আবহাওয়া অফিস বলছে, গত কয়েকদিন ধরে আলোচনার কেন্দ্রে রয়েছে তীব্র গরমের কষ্ট।
Summary Generated by mBART	বাংলাদেশের আবহাওয়া অফিস বলছে, গত ৫০ বছরে ঢাকায় সবচাইতে বেশি গরম পড়েছে এবার। আর তাতেই হঠা□ করে বেড়ে গেছে শীতাতপ নিয়ন্ত্রণের যন্ত্র বা এসি বিক্রি। বিক্রেতারা বলছেন, এসির চাহিদা এতই বেশি যে তারা সরবরাহ করতে পারছেন না। অন্যদিকে এসির ঠান্ডা হাওয়া উপভোগ করতে বাড়ছে বিদ্যুতের ব্যবহার। ফলে এই গরমে তা আবার বাড়িয়ে দিচ্ছে লোডশেডিং।
Summary Generated by BanglaT5	ঢাকায় অস্বাভাবিক গরমের তীব্রতায় বাড়ছে এসির চাহিদা।
Summary Generated by Llama 3.1	গত কিছুদিন ধরে ঢাকার বাসিন্দাদের আলোচনার কেন্দ্রে রয়েছে অসহনীয় গরম, যার কারণে এয়ার কন্ডিশনারের চাহিদা উল্লেখযোগ্যভাবে বৃদ্ধি পাচ্ছে।

Summary Generated by Bloom-576M	আবহাওয়া অফিস বলছে, গত ৫০ বছরে ঢাকায় সবচাইতে বেশি গরম পড়েছে এবার। আর তাতেই হঠা□ করে বেড়ে গেছে শীতাতপ নিয়ন্ত্রণের যন্ত্র বা এসি বিক্রি। বিক্রেতারা বলছেন, এসির চাহিদা এতই বেশি যে তারা সরবরাহ করতে পারছেন না। কিছু দোকান একবারেই খালি। বিক্রেতারা বলছেন, এক বা দেড়টনের এসি - যেটা সাধারণ মাপের ঘরের জন্য দরকার - তা বাজারে প্রায় নেই। পাওয়া যাচ্ছে শুধু দুই টনের এসি। ঢাকার একটি বিপণী বিতানে এসি কিনবেন বলে এসেছেন ব্যবসায়ী মোহাম্মদ হানিফ। তিনি বলছেন, বেশি চাহিদার কারণে দামও বাড়িয়ে দিয়েছেন বিক্রেতারা। অন্যদিকে এসির ঠান্ডা হাওয়া উপভোগ করতে বাড়ছে বিদ্যুতের ব্যবহার। ফলে এই গরমে তা আবার বাড়িয়ে দিচ্ছে লোডশেডিং।

Here we can see the summary generated by mBART is significantly longer than the reference summary. And the length of the other generated summaries is within an acceptable range. Moreover, Bloom generated detailed passage instead of concise summary.

Original Summary	আবহাওয়া অধিদপ্তরের বিশেষ বিজ্ঞপ্তিতে জানানো হয়েছে যে সুপার সাইক্লোন আম্পান আজ (মঙ্গলবার) বেলা বারটায় চট্টগ্রাম সমুদ্রবন্দর থেকে ৮১০ কিলোমিটার, মংলা থেকে ৬৯৫ ও পায়রা সমুদ্র বন্দর থেকে ৬৯০ কিলোমিটার দক্ষিণ-দক্ষিণ পশ্চিমে অবস্থান করছিলো।
Summary Generated by mT5	ঘূর্ণিঝড় আম্পান বাংলাদেশের উপকূলের দিকে ঢুকে পড়তে পারে বলে আবহাওয়া অধিদপ্তর জানিয়েছে। ঘূর্ণিঝড় আম্পান যখন আঘাত হানবে তা অতি প্রবল হওয়ার যথেষ্ট আশঙ্কা রয়েছে।
Summary Generated by mBART	বাংলাদেশের উপকূল অতিক্রম করতে পারে ঘূর্ণিঝড় আম্পান। এটিকে বলা হচ্ছে ’৭ নম্বর বিপদসংকেত মহামারির মধ্যে আঘাত হানবে,’ বলছেন আবহাওয়া অধিদপ্তরের কর্মকর্তা ও গবেষকরা।
Summary Generated by BanglaT5	বাংলাদেশে ’অতি প্রবল’ ঘূর্ণিঝড় আম্পান আঘাত হানা দিতে পারে বলে আশঙ্কা করা হচ্ছে।
Summary Generated by Llama 3.1	ঘূর্ণিঝড় আম্পান যখন আঘাত হানবে তা অতি প্রবল হওয়ার যথেষ্ট আশঙ্কা রয়েছে।

Summary Generated by Bloom-576M	স্যাটেলাইট চিত্রে ঘূর্ণিঝড় আম্পানের বর্তমান অবস্থান এটি উত্তর উত্তরপূর্ব দিকে অগ্রসর হয়ে খুলনা ও চট্টগ্রামের মধ্যবর্তী অঞ্চল দিয়ে কাল ভোররাত থেকে বিকেল বা সন্ধ্যার মধ্যে বাংলাদেশ উপকূল অতিক্রম করতে পারে। আবহাওয়াবিদ বজলুর রশিদ বিবিসি বাংলাকে জানিয়েছেন, মূলত বাংলাদেশ ও ভারতের সীমান্তের সুন্দরবন অংশে ঘূর্ণিঝড়টির মূল অংশ আঘাত হানতে পারে। "ঝড়ের মূল অংশ সুন্দরবন অংশে এলেও এর প্রভাব পড়বে চারদিকেই। তবে এখনো এটি এক হাজার কিলোমিটারেরও বেশি দূরে। তাই নানা পরিবর্তনের সুযোগ রয়েছে," বলছিলেন মিস্টার রশিদ। তার মতে, ঝড়টি ভারতের দীঘা থেকে বাংলাদেশের সন্দ্বীপ এলাকার মধ্য দিয়ে যাবে এবং এর মূল অংশ ভারত বাংলাদেশ সীমান্তের সুন্দরবন অংশে আসবে। আবহাওয়া অধিদপ্তরের বিজ্ঞপ্তিতে আরও বলা হয়েছে যে ঘূর্ণিঝড় ও অমাবস্যার প্রভাবে উপকূলীয় অঞ্চলে ৫-১০ ফুট উচ্চতার জলোচ্ছ্বাস হতে পারে। আবহাওয়া অধিদপ্তরের পরিচালক সামছুদ্দিন আহমেদ বলছেন, ঘূর্ণিঝড় আম্পান যখন আঘাত হানবে তা অতি প্রবল হওয়ার যথেষ্ট আশঙ্কা রয়েছে।

From this example it is visible that the summary produced by Bangla T5 and Llama 3.1 is noticeably shorter than the reference summary.

- Because of the presence of union words and complex structures, few summaries ended up with meaningless messages.

  Example:

Original Summary	বিশ্বে এখন আলোচিত ব্যক্তি যুক্তরাষ্ট্রের প্রেসিডেন্ট ডোনাল্ড ট্রাম্প। নিজের কাজ এবং কথার জন্য তিনি গণমাধ্যমের আলোচনার কেন্দ্রবিন্দুতেও পরিণত হয়েছেন।
Summary Generated by mT5	যুক্তরাষ্ট্রের প্রেসিডেন্ট নির্বাচিত ডোনাল্ড ট্রাম্পকে নিয়ে সোশ্যাল মিডিয়ায় ব্যাপক আলোচনা চলছে।
Summary Generated by mBART	হোয়াইট হাউজে ডোনাল্ডট্রাম্পকে দেখানোর পর গত কয়েকদিন ধরে সামাজিক যোগাযোগ মাধ্যমে ব্যাপকভাবে ছড়িয়ে পড়েছে নানা ধরণের কৌতুক, এমনকি হাস্যরসাত্মক ভিডিও -যেছে তাকে খুবই ক্ষুদ্র ও বিরল।
Summary Generated by BanglaT5	বিশ্বজুড়ে বর্তমানে আলোচনার কেন্দ্রবিন্দু হলেন যুক্তরাষ্ট্রের প্রেসিডেন্ট ডোনাল্ডট্রাম্প।
Summary Generated by Llama 3.1	বারাক ওবামা একটি সোফায় বসে আছেন।

Summary Generated by Bloom-576M	সাম্প্রতিক খবরের ছবিগুলো নিয়েই ফটোশপ করা হয়েছে ট্রাম্পকে। মি: ট্রাম্পের বিরোধীরা তাঁর হাতের সাইজ নিয়ে মন্তব্য করলে এর প্রতিক্রিয়াও জানিয়েছেন ট্রাম্প। যুক্তরাষ্ট্রের প্রেসিডেন্ট নির্বাচিত হবার পর থেকেই ট্রাম্পকে নিয়ে নানা কৌতুক বানানো হচ্ছে, এমনকি তাঁর প্রশাসনের কজনকে নিয়ে বানানো হয়েছে হাস্যরসাত্মক ভিডিও -যেগুলো জনপ্রিয়তাও পাচ্ছে। তবে এখন কিছু ইন্টারনেট ব্যবহারকারী প্রেসিডেন্ট ট্রাম্পকে দিয়েছে নতুন রূপ। মি: ট্রাম্পের আকৃতি একদম ক্ষুদ্র বানিয়ে প্রকাশ করা হয়েছে বেশ কয়েকটি ছবি, ইন্টারনেটে যা ছড়িয়ে পড়েছে। প্রায় ৬ ফুট ২ ইঞ্চি উচ্চতাসম্পন্ন ডোনাল্ড ট্রাম্প বিশ্বের অনেক নেতাদের তুলনায় লম্বা, এমনকি বারাক ওবামার থেকেও ট্রাম্প এক ইঞ্চি বেশি লম্বা। কিন্তু যে ছবিগুলো প্রকাশ পেয়েছে তাতে ট্রাম্পকে অন্যদের তুলনায় ক্ষুদ্রাকৃতির ব্যক্তি হিসেবে দেখা যাচ্ছে। মি: ট্রাম্পের অফিশিয়াল এ ছবিগুলো ফটোশপের মাধ্যমে পরিবর্তন করে রেডিট ব্যবহারকারীরা চাইছে প্রেসিডেন্টের প্রতিক্রিয়া। তবে এখন পর্যন্ত ডোনাল্ড ট্রাম্পের কোনও সাড়া পাওয়া যায়নি।

As we can observe, the summary produced by Llama 3.1 concludes with an irrelevant message.

- Several summaries added unnecessary content.
  Example:

Original Summary	আদ্দিস আবাবার উঁচু অফিস ভবনে যে ডেস্কে বসে মারকোস লেমা কাজ করেন, সেখান থেকে পুরো নগরীর একটা মনোরম দৃশ্য দেখা যায়।
Summary Generated by mT5	সৌদি আরবের ইন্টারনেট বন্ধ করে দেয়ার ঘটনা ঘটেছে গত বছর।
Summary Generated by mBART	বিশ্বের বিভিন্ন দেশে ইন্টারনেট ব্ল্যাকআউট করা হয়েছে এবং এর প্রতিক্রিয়া বিশ্বের বিভিন্ন দেশে বিভিন্ন রকম ছিল।
Summary Generated by BanglaT5	আদ্দিস আবাবার উঁচু অফিস ভবনে বসে পুরো শহরের একটি দৃষ্টিনন্দন দৃশ্য দেখতে পান।
Summary Generated by Llama 3.1	উঁচু অফিস ভবন থেকে শহরের মনোরম দৃশ্য স্পষ্টভাবে দেখা যায়।

Summary Generated by Bloom-576M	ইন্টারনেট ব্ল্যাকআউট: নানা দেশে নানা অজুহাতে সরকার এই কাজ করে মারকোস লেমা একটি প্রযুক্তি উদ্ভাবন কোম্পানি 'আইসএডিস' এর প্রতিষ্ঠাতা। তার এই কাজের জায়গাটি প্রতিদিন যাদের উপস্থিতিতে সরগম থাকে, তাদের অনেকের চোখেই বড় স্বপ্ন। কিন্তু যখন ইন্টারনেট বন্ধ করে দেয়া হয়, তখন তাদের সবকিছু অংকুরেই যেন নষ্ট হয়ে যায়। একটি ডিজিটাল অধিকার গোষ্ঠী 'একসেস নাউ' বিবিসির সঙ্গে যেসব তথ্য শেয়ার করেছে, তাতে দেখা যাচ্ছে গত বছর ৩৩টি দেশে অন্তত ২০০ বার ইচ্ছেকৃতভাবে ইন্টারনেট বন্ধ করে দেয়া হয়েছিল। "যখন এমনটা হয়, তখন এখানে গাড়ি চলাচলও যেন বন্ধ হয়ে যায়। এখানে কেউ আসে না। যদিও আসে, বেশিক্ষণ থাকে না। কারণ ইন্টারনেট ছাড়া তারা এখানে করবেটাই বা কী", বলছেন মারকোস। "একটা সফটওয়্যার তৈরি করে দেয়ার জন্য আমরা একটা চুক্তি করেছিলাম, কিন্তু আমাদের সেটি বাতিল করে দিতে হয়েছিল, কারণ আমরা সময়মত কাজটা করতে পারতাম না। এখানে কোন ইন্টারনেট সংযোগ ছিল না। আমাদের আন্তর্জাতিক কাস্টমাররা মনে করতো, আমরা তাদের উপেক্ষা করছি। আসলে ব্যাপারটা তা নয়, আমাদের কিছু করার ছিল না।" আদ্দিস আবাবায় নিজের অফিসে মারকোস লেমা "মোটরবাইক চালকরা খাবার পৌঁছে দেয়ার বদলে বসে থাকে। ইন্টারনেট সংযোগ ছাড়া মানুষ তো অনলাইনে বা অ্যাপের মাধ্যমে অর্ডার দিতে পারে না", বলছেন মারকোস।

From this example we can see the summaries generated by mBART and mT5 include unnecessary content. However, the summaries produced by the other models provide relevant information. And the passage generated by bloom is also not relevant.

- Important information was missing in several summaries.

  Example:

Original Summary	সিঙ্গাপুরে হাসপাতালে ভর্তি অবস্থায় মারা গেছেন একুশে পদকপ্রাপ্ত জনপ্রিয় সংগীতশিল্পী সুবীর নন্দী।
Summary Generated by mT5	বাংলাদেশের সুপরিচিত লোক সুবীর নন্দী আজ মারা গেছেন।
Summary Generated by mBART	সুবীর নন্দী সিঙ্গাপুরে মৃত্যুবরণ করেছেন।
Summary Generated by BanglaT5	বাংলাদেশের জনপ্রিয় একটি গানের শিল্পী সুবীর নন্দী মারা গেছেন আজ।
Summary Generated by Llama 3.1	সুবীর নন্দী বাংলাদেশের সবচেয়ে বিখ্যাত গায়ক ছিলেন।

Summary Generated by Bloom-576M	সুবীর নন্দী জাতীয় বার্ন অ্যান্ড প্লাস্টিক সার্জারি ইন্সটিটিউটের জাতীয় সমন্বয়ক অধ্যাপক সামন্ত লাল সেন বিবিসি বাংলাকে জানিয়েছেন, বাংলাদেশ সময় আজ ভোর সাড়ে চারটায় সুবীর নন্দী মারা গেছেন। অধ্যাপক সেন জানিয়েছেন, সিঙ্গাপুরে চিকি□সাধীন অবস্থায় পরপর তিনবার হৃদরোগে আক্রান্ত হয়েছিলেন সুবীর নন্দী। এর আগে ঢাকায় চিকি□সাধীন অবস্থায়ও একবার হার্ট অ্যাটাক হয়েছিল তার। সেসময় ১৮ দিন ঢাকার সম্মিলিত সামরিক হাসপাতালে চিকি□সাধীন থাকার পর গত ৩০শে এপ্রিল সিঙ্গাপুরে নেওয়া হয় সুবীর নন্দীকে। বাংলাদেশে সুবীর নন্দীর চিকি□সার বিষয়টি সমন্বয় করছিলেন অধ্যাপক সেন। তিনি জানিয়েছেন, "বারবার হার্ট অ্যাটাক হয়েছে তার, সেই সঙ্গে সুবীর নন্দীর মাল্টিপল অরগ্যান ফেইলিউর হচ্ছিল।" সঙ্গীতে অবদানের জন্য এ বছরই তাকে একুশে পদকে ভূষিত করে সরকার। ১৯৭২ সালে ঢাকা রেডিওতে প্রথম রেকর্ডিং এর মধ্য দিয়ে গানের জগতে আসেন সুবীর নন্দী। ৪০ বছরের দীর্ঘ ক্যারিয়ারে গেয়েছেন আড়াই হাজারেরও বেশি গান। সিলেটে থাকতে শিখেছিলেন নজরুল সঙ্গীত।

This example shows that Llama 3.1 could not produce important information.

- There were some incomplete summaries.

  Example:

Original Summary	বাংলাদেশে যে কোন ব্যাংক সাইবার হামলার শিকার হতে পারে বলে সতর্ক করে দিয়েছে সরকারের অর্থ মন্ত্রণালয়। সতর্কতা হিসেবে ইতিমধ্যেই নিরাপত্তা বাড়ানোর কথা জানিয়েছে ব্যাংকগুলো।
Summary Generated by mT5	বাংলাদেশে ব্যাংকগুলোর এটিএম বুথ থেকে টাকা উত্তোলন ও সুইফট নেটওয়ার্কে হামলার আশংকা প্রকাশ করার জন্য সতর্কতামূলক ব্যবস্থা নিতে বলা হয়েছে।
Summary Generated by mBART	গত আড়াই মাসের মধ্যে সবচেয়ে বড় রিজার্ভ চুরির ঘটনার পর বেশ কিছু রাষ্ট্রায়ত্ত এবং বাণিজ্যিক ব্যাংকগুলোকে এটিএম বুথের কার্যক্রম সীমিত করার পর এ বিষয়ে সতর্কতা জানিয়ে বলেছে রাষ্ট্রীয় মন্ত্রণালয়ের টাকা উত্তোলন।
Summary Generated by BanglaT5	বাংলাদেশ ব্যাংকের সতর্কতা
Summary Generated by Llama 3.1	বাংলাদেশে কেন্দ্রীয় ব্যাংকের সতর্কতা জারির পর আজ (বৃহস্পতিবার) ব্যাংকগুলোও সতর্কতা জারি করা হয়েছে।

Summary Generated by Bloom-576M	বাংলাদেশের রিজার্ভ চুরির ঘটনা সারা বিশ্বে তোলপাড় সৃষ্টি করেছিল। সেই সঙ্গে কয়েকটি রাষ্ট্রায়ত্ত এবং বাণিজ্যিক ব্যাংক তাদের এটিএম বুথের কার্যক্রম সীমিত করেছে। ২০১৬ সালে সাইবার হামলার মাধ্যমে বাংলাদেশের কেন্দ্রীয় ব্যাংকের মোট ১০ কোটি ১০ লক্ষ ডলারের রিজার্ভ চুরির ঘটনা ঘটেছিল, যাকে ব্যাংকিং খাতে পৃথিবীতে এখনো পর্যন্ত সবচেয়ে বড় চুরির ঘটনা বলা হয়। এরপর থেকেই সাইবার হামলার বিষয়টি দেশের ব্যাংকিং খাতের জন্য একটি উদ্বেগের বিষয়। অর্থ মন্ত্রণালয়ের চিঠি সাইবার হামলার আশংকায় গত আড়াই মাসের মধ্যে দ্বিতীয়বারের মত সতর্ক করা হয়েছে দেশের সরকারি-বেসরকারি দুই ধরনের ব্যাংককে। অর্থ মন্ত্রণালয়ের আর্থিক প্রতিষ্ঠান বিভাগ থেকে দেশের কেন্দ্রীয় ব্যাংককে চিঠি দিয়ে জানানো হয়, ২০১৬ সালে বাংলাদেশ ব্যাংকের রিজার্ভ থেকে অর্থ চুরির সঙ্গে জড়িত উত্তর কোরিয়া-ভিত্তিক হ্যাকার গ্রুপ আবারো বাংলাদেশের বিভিন্ন ব্যাংকে সাইবার হামলা চালাতে পারে। বাংলাদেশে এটিএম বুথে জালিয়াতির ঘটনা নতুন নয় এক্ষেত্রে ব্যাংকগুলোর এটিএম বুথ থেকে টাকা উত্তোলন ও সুইফট নেটওয়ার্কে হামলা হবার আশংকা প্রকাশ করা হয়েছে। এ বিষয়ে ব্যাংকগুলোকে প্রয়োজনীয় সতর্কতামূলক ব্যবস্থা নিতে বলা হয় চিঠিতে। অর্থ মন্ত্রণালয়ের একজন দায়িত্বশীল কর্মকর্তা বিবিসিকে জানিয়েছেন, মন্ত্রণালয় থেকে বৃহস্পতিবার কেন্দ্রীয় ব্যাংক এবং রাষ্ট্রায়ত্ত ব্যাংকগুলোতে ওই চিঠি পাঠানো হয়। এরপর কেন্দ্রীয় ব্যাংক বাণিজ্যিক ব্যাংকগুলোতেও এ সংক্রান্ত সতর্কতা পাঠিয়েছে।

Here, we can notice that Bangla T5 generated an unfinished summary.

- **R**epetition words were generated in a few summaries.

  Example:

Original Summary	ভারতের রাজধানী দিল্লির একটি জনবহুল এলাকায় হোটেলে আগুন লেগে অন্তত ১৭ জনের মৃত্যু হয়েছে। মৃতদের মধ্যে এক নারী ও একটি শিশুও রয়েছে।
Summary Generated by mT5	দিল্লির একটি জনবহুল এলাকায় হোটেলে আগুন লেগে অন্তত ১৭ জনের মৃত্যু হয়েছে।
Summary Generated by mBART	ভারতের রাজধানী দিল্লিতে একটি জনবহুল এলাকায় একটি হোটেলে আগুন লেগে অন্তত ১৭ জনের মৃত্যু হয়েছে, যাদের মধ্যে বেশ কয়েকজন আবাসিকও রয়েছেন বলে জানাচ্ছেন নিপরিচিত তল্লাশী দমকল কর্মী ও দরিদ্র কর্মীরা।
Summary Generated by BanglaT5	ভারতে রাজধানী দিল্লির একটি বিলাসবহুল হোটেল 'মন্দিরবাগে' আজ ভোররাতে আগুন লাগে।
Summary Generated by Llama 3.1	দিল্লির একটি জনবহুল এলাকায় হোটেলে আগুন আগুন লেগে অন্তত ১৭ জনের মৃত্যু হয়েছে। হয়েছে।

Summary Generated by Bloom-576M	ভারতের রাজধানী দিল্লির একটি জনবহুল এলাকায় হোটেলে আগুন লেগে অন্তত ১৭ জনের মৃত্যু হয়েছে। নিহতরা আগুনের গ্রাস থেকে বাঁচতে হোটেলের জানালা থেকে ঝাঁপ দিয়েছিলেন বলে পুলিশ জানাচ্ছে। আরও বেশ কয়েকজন জানালা আর হোটেলের ছাদ থেকে ঝাঁপ দিয়ে প্রাণে বেঁচেছেন। তবে দিল্লি সরকারের মন্ত্রী সত্যেন্দ্র জৈন বলছেন, বেশির ভাগ মৃত্যুই হয়েছে দমবন্ধ হয়ে। জানা যাচ্ছে, পর্যটকরা যে এলাকার হোটেলে থাকতে পছন্দ করেন, সেরকমই একটি এলাকা, কারোল বাগের 'হোটেল অর্পিত প্যালেস' এ আজ ভোর রাতে আগুন লাগে। দমকল অধিকর্তা জি সি মিশ্র সংবাদ মাধ্যমে জানিয়েছেন, ভোর সাড়ে চারটের দিকে তাদের কাছে আগুন লাগার খবর আসে। চারতলায় প্রথমে আগুন লেগেছিল, পরে তা তিনতলা আর দোতলাতে ছড়িয়ে পড়ে। ২০টি দমকল গাড়ি কয়েক ঘন্টার চেষ্টার পরে আগুন নিয়ন্ত্রণে আনতে পেরেছে বলে জানা গেছে। দমকল কর্মীরা হোটেল থেকে ৩৫ জনকে উদ্ধার করতে পেরেছেন। আগুন লাগার সময়ে হোটেলের ৪০ টি ঘরে ৬০ জন আবাসিক আবাসিক আবাসিকছিলেন, যাদের বেশিরভাগই বেশিরভাগই বেশিরভাগই বেশিরভাগই তখন ঘুমিয়ে ছিলেন বলে দমকল কর্মকর্তারা জানাচ্ছেন। দমকল কর্মীরা হোটেল থেকে ৩৫ জনকে উদ্ধার করতে পেরেছেন। হোটেলের করিডোরগুলিতে কাঠের প্যানেলিং থাকায় আগুন দ্রুত ছড়িয়ে পড়ে, আর আবাসিকরাও করিডোর ধরে বেরিয়ে আসতে পারেন নি।

Llama 3.1 produced some redundant words in summary. Likewise, there are repetitive words in the passage produced by Bloom too.

Chapter 8

Limitation and Future Work

The present state of Bangla text summarization is not favorable due to certain constraints. For instance, there is a vast amount of data accessible for the English language since it is a commonly used language. However, there is a scarcity of data regarding Bangla language. Furthermore, the Bangla language has a complicated script. The grammatical intricacies, idiomatic expressions and complicated sentence patterns are difficult to summarize. Additionally, there is a shortage of adequate resources. A lack of resources can affect the standard of models for the Bangla language. Furthermore, Bangla provides a wide range of different kinds of content. Handling various types of content can be challenging. Moreover, there were some computational challenges during this research work. GPU availability was one of them, which restricted this research from running larger-scale experiments efficiently.

In the future, our plan is to improve the Bangla text summarization by expanding our dataset from more diverse domains to ensure a large area of coverage. Also, we will work on manual annotation on our data to ensure the best quality of our dataset. With better computational resources, we aim to fine-tune advanced models like Llama specially for Bangla text summarization, which was not feasible in this work due to time constraints and computational limitations. Moreover, we want to explore the performance of more models to evaluate their performance and ensure the best possible results in our research area.

Chapter 9

Conclusion

Enhancing Bengali text summarization requires a comprehensive strategy that involves comprehending the language, developing a framework, organizing the data, and assessing its effectiveness. To create a concise summary, it is necessary to have a thorough comprehension of the language and context of the text. In our research. Ongoing investigation in the field of NLP aids in the development of progress in summarizing methods. Summarizing Bengali language content is a significant and difficult goal. In our research, we have collected the XL-Sum dataset from BUET CSE NLP Group and also extended the dataset by fetching text-summary pair data from The Daily Star newspaper to include different types of domains. Different domains help explore models differently. Then we pre processed our dataset which we have fetched. We have fine-tuned mT5, BanglaT5, mBART and Bloom-576M models. We also used Llama 3.1 but could not train the model Due to limited computational resources. Then we have analyzed the results across different models with ROUGE metric (ROUGE-1, ROUGE-2 and ROUGE-L) and BERTScore metric (to check semantic similarity). To find out the quality of the summaries generated by the models, we also used Abstractivity, Compression, Novel n-grams, Redundancy matrices as it is difficult to understand the semantic similarity of abstractive summary by ROUGE metric. After evaluating the results, we have seen the quality of summary of BanglaT5, mBART and Llama are close to the quality of summary of Mt5. There are some limitations we had to face. Bangla is a low resource language due to the relatively limited availability of datasets, linguistic resources and tools compared to high resource languages like English. Also we had limited computational capabilities. In future the scores of Llama 3.1 can be better if we get more computational capacity. Also we will also increase the dataset from different domains and try more advanced models. Looking at different models helps us understand how they work and the problems caused by limited computing power. Our research shows the difficulties and possibilities of natural language processing for languages like Bengali, which have limited data and complex word structures.

Bibliography

[1] K. Papinesi, ?Bleu: A method for automatic evaluation of machine translation,? **in***Proc. 40th Actual Meeting of the Association for Computational Linguistics (ACL), 2002* 2002, **pages** 311–318.

[2] M. N. Uddin **and** S. A. Khan, ?A study on text summarization techniques and implement few of them for Bangla language,? **in***2007 10th international conference on computer and information technology* IEEE, 2007, **pages** 1–4.

[3] N. Begum, M. A. Fattah **and** F. Ren, ?Automatic text summarization using support vector machine,? *international journal of innovative computing information and control*, **jourvol** 5, **number** 7, **pages** 1987–1996, 2009.

[4] L. Suanmali, N. Salim **and** M. S. Binwahlan, ?Fuzzy logic based method for improving text summarization,? *arXiv preprint arXiv:0906.4690*, 2009.

[5] M. I. A. Efat, M. Ibrahim **and** H. Kayesh, ?Automated Bangla text summarization by sentence scoring and ranking,? **in***2013 International Conference on Informatics, Electronics and Vision (ICIEV)* IEEE, 2013, **pages** 1–5.

[6] K. Sarkar, ?A keyphrase-based approach to text summarization for English and bengali documents,? *International Journal of Technology Diffusion (IJTD)*, **jourvol** 5, **number** 2, **pages** 28–38, 2014.

[7] J.-P. Ng **and** V. Abrecht, ?Better summarization evaluation with word embeddings for ROUGE,? *arXiv preprint arXiv:1508.06034*, 2015.

[8] A. M. Rush, S. Chopra **and** J. Weston, ?A neural attention model for abstractive sentence summarization,? *arXiv preprint arXiv:1509.00685*, 2015.

[9] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang **andothers**, ?Abstractive text summarization using sequence-to-sequence rnns and beyond,? *arXiv preprint arXiv:1602.06023*, 2016.

[10] S. Akter, A. S. Asa, M. P. Uddin, M. D. Hossain, S. K. Roy **and** M. I. Afjal, ?An extractive text summarization technique for Bengali document (s) using K-means clustering algorithm,? **in***2017 ieee international conference on imaging, vision & pattern recognition (icivpr)* IEEE, 2017, **pages** 1–6.

[11] M. Grusky, M. Naaman **and** Y. Artzi, ?Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies,? *arXiv preprint arXiv:1804.11283*, 2018.

[12] S. Narayan, S. B. Cohen **and** M. Lapata, ?Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization,? *arXiv preprint arXiv:1808.08745*, 2018.

[13] A. Rahman, F. M. Rafiq, R. Saha **and** R. Rafian, ?Bengali text summarization using TextRank, Fuzzy C-means and aggregated scoring techniques,? phdthesis, BRAC University, 2018.

[14] L. Wang, J. Yao, Y. Tao, L. Zhong, W. Liu **and** Q. Du, ?A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization,? *arXiv preprint arXiv:1805.03616*, 2018.

[15] M. Lewis, ?Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,? *arXiv preprint arXiv:1910.13461*, 2019.

[16] Y. Liu **and** M. Lapata, ?Text summarization with pretrained encoders,? *arXiv preprint arXiv:1908.08345*, 2019.

[17] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger **and** Y. Artzi, ?Bertscore: Evaluating text generation with bert,? *arXiv preprint arXiv:1904.09675*, 2019.

[18] R. Bommasani **and** C. Cardie, ?Intrinsic evaluation of summarization datasets,? **in***Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 2020, **pages** 8075–8096.

[19] Y. Liu, ?Multilingual denoising pre-training for neural machine translation,? *arXiv preprint arXiv:2001.08210*, 2020.

[20] C. Raffel, N. Shazeer, A. Roberts **andothers**, ?Exploring the limits of transfer learning with a unified text-to-text transformer,? *The Journal of Machine Learning Research*, **jourvol** 21, **number** 1, **pages** 5485–5551, 2020.

[21] J. Zhang, Y. Zhao, M. Saleh **and** P. Liu, ?Pegasus: Pre-training with extracted gap-sentences for abstractive summarization,? **in***International Conference on Machine Learning* PMLR, 2020, **pages** 11 328–11 339.

[22] N. Dhar, G. Saha, P. Bhattacharjee, A. Mallick **and** M. S. Islam, ?Pointer over attention: An improved bangla text summarization approach using hybrid pointer generator network,? **in***2021 24th International Conference on Computer and Information Technology (ICCIT)* IEEE, 2021, **pages** 1–5.

[23] T. Hasan, A. Bhattacharjee, M. S. Islam **andothers**, ?XL-sum: Large-scale multilingual abstractive summarization for 44 languages,? *arXiv preprint arXiv:2106.13822*, 2021.

[24] S. Xu, X. Zhang, Y. Wu **and** F. Wei, ?Sequence level contrastive learning for text summarization,? **in***Proceedings of the AAAI conference on artificial intelligence* **volume** 36, 2022, **pages** 11 556–11 565.

[25] S. A. I. Hayat, A. Das **and** M. M. Hoque, ?Abstractive Bengali Text Summarization Using Transformer-based Learning,? **in***2023 6th International Conference on Electrical Information and Communication Technology (EICT)* IEEE, 2023, **pages** 1–6.

[26] T. Le Scao, A. Fan, C. Akiki **andothers**, ?Bloom: A 176b-parameter open-access multilingual language model,? 2023.

[27] I. Ni'mah, M. Fang, V. Menkovski **and** M. Pechenizkiy, ?Nlg evaluation metrics beyond correlation analysis: An empirical metric preference checklist,? *arXiv preprint arXiv:2305.08566*, 2023.

[28] M. Wang, P. Xie, Y. Du **and** X. Hu, ?T5-Based Model for Abstractive Summarization: A Semi-Supervised Learning Approach with Consistency Loss Functions,? *Applied Sciences*, **jourvol** 13, **number** 12, **page** 7111, 2023.

[29] N. Molleti, *Understanding BART: A Breakdown of the Model for Natural Language Processing*, https://www.linkedin.com/pulse/understanding-bart-breakdown-model-natural-language-nagababu-molleti-swtxc/, Accessed: 15-Oct-2024, 2024.

[30] R. Vavekanand **and** K. Sam, *Llama 3.1: An In-Depth Analysis of the Next-Generation Large Language Model*, 2024.