Enhancing Bangla Text Summarization through Transformer Learning and Ranking-Based Approaches

Submitted by

Md. Tousif Rahman

ID: CSE 02207162

Batch: CSE 22 A Day

Under The Supervision of

Tahmina Akter

Senior Lecturer,

Department of Computer Science and Engineering

Port City International University

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

Port City International University

7-14, Nikunja Housing Society, South Khulshi

Chattogram, Bangladesh

February, 2025

DECLARATION

We respectfully affirm that the work for our undergraduate degree **"Enhancing Bangla Text Summarization through Transformer Learning and Ranking-Based Approaches"** is entirely original. This thesis includes correctly cited sections throughout.

Md. Tousif Rahman

Student ID: CSE 02207162

Batch: CSE 22 A Day

Department of Computer Science and Engineering

Port City International University

APPROVAL FOR SUBMISSION

This thesis titled **"Enhancing Bangla Text Summarization through Transformer Learning and Ranking-Based Approaches"** by **Md. Tousif Rahman** Student ID: CSE 02207162. It has been authorized for to be submitted to Port City International University's Department of Computer Science and Engineering in partial fulfillment of the criteria for the Degree of Bachelor of Science.

(Signature of the supervisor)

Tahmina Akter
Senior Lecturer,
Department of Computer Science and Engineering
Port City International University
Email: tahminashopna499@gmail.com

Dedication

"Dedicated to my cherished parents and respectable teachers".

ACKNOWLEDGEMENT

I initiate this acknowledgement by expressing my deepest gratitude to those who have supported me throughout this academic journey. Foremost, I extend my heartfelt thanks to the Almighty for the wisdom and perseverance bestowed upon me. I am profoundly indebted to my respected supervisor, Tahmina Akter, whose guidance and expertise have been the cornerstone of my research. Her unwavering support and insightful critiques have been invaluable. Additionally, I am grateful to the teachers, seniors and friends, whose encouragement and insights have been a constant source of strength and motivation. Their collective wisdom has been a guiding light in this academic endeavor.

Table of Contents

ABSTRACT

Thus nowadays, the requirements for text summarization techniques that are both efficient and accurate make it a necessity to look for ways to enhance the quality and accuracy of pre-trained models designed for summarizing Bengali texts. For text summarization tasks, there are several options among numerous pre-trained transformer models. From these summaries generated by pre-trained summarization models, it becomes quite challenging to get the one that is most informative and relevant to the given text. This paper seeks to use a simple yet effective ranking approach via TextRank to compare the outputs of four different pre-trained Bengali text summarizers in order to identify the most accurate and most informative summary of a given text. The procedure starts with preprocessing of the input text, which means putting out unwanted elements such as special characters and punctuation marks. After that, summaries are generated from the input texts using four different pre-trained summarizing models, and ranking will be applied to find the most suitable summary. The final summary will be chosen as that with the highest-ranking score. To assess the effectiveness of this approach, generated summaries will be compared to human-annotated summaries using standard NLG metrics such as BLEU, ROUGE, BERTScore, WIL, WER, and METEOR. Experimental results affirm that the combination of sorts by ranking, bringing together the strengths of every pre-trained transformer model, greatly enhances the efficacy and accuracy of Bengali text summarization.

Keywords: Text summarization, pre-trained, transformer models, TextRank, human-annotated, NLG metrics, BLEU, ROUGE, BERTScore, WIL, WER, METEOR.

CHAPTER 1

INTRODUCTION

1.1 Overview

Since the tremendous increase in digital contents is witnessed, automatic summarization started gaining its importance due to the need and aspiration of summarizing just the principal information that retains substantial sense whole from whatever has been authored with respect to length-duration and effort [3], although as resource-rich languages in such aspects-state-of-the-art performance is seen lacking in the case of the Bengali language so far [2].

Bengali, having over 250 million native speakers worldwide, has recently garnered huge interest in NLP research works [3]. However, it is still difficult to develop successful abstractive summarization models for Bengali due to the unavailability of linguistic resources and computationally expensive model training processes [8]. Classical methods of text summarization for Bengali texts have mostly been concentrated on extractive techniques, i.e., extracting significant sentences from the text instead of generating novel, human-like summaries [2]. Recent developments in transformer-based models, i.e., T5, BERT, and mT5, have greatly enhanced the state of the art in text summarization tasks [3].

One of the key issues for summarizing Bengali texts is identifying the best and most informative summary out of a pool of candidate outputs of several models [1]. The groundbreaking paper, "Rank Your Summaries: Improving Bengali Text Summarization through Ranking-Based Approach", proposes a new ranking-based approach to overcome this issue [1]. This method utilizes multiple pre-trained models for summarizing Bengali text, involves a ranking algorithm, and identifies the most informative and contextually relevant summary [1]. This method incorporates both extractive and abstractive summarization and hence offers far greater accuracy and coherence compared to conventional methods [1].

Besides, XL-Sum has also contributed a lot to multilingual abstractive summarization.

High-quality summaries annotated by professionals are provided in 44 languages, including Bengali[1]. This is a helpful dataset for training Bengali summarization models, which provide competitive performance despite low resource settings, and performance is excellent with higher-than-expected ROUGE scores on both metrics with Transformer-based multilingual models like mT5 fine-tuned on XL-Sum for Bengali text summarization [5].

The assessment of summarization is still a crucial field of study. Conventional metrics, like Word Error Rate, have substantial limitations in their ability to quantify text quality, since they focus almost exclusively on edit distance and not on semantic precision [15]. The comparison of WER and RIL against MER and WIL introduces the concepts of Match Error Rate and Word Information Lost, which provide a deeper insight into summarization performance by quantifying the level of information preserved or lost [15]. These new metrics can be incorporated into Bengali summarization tests to offer an improved measure of summary quality beyond the traditional word-level modifications [15].

Against this backdrop, this research further advances the domain of Bengali text summarization using state-of-the-art transformer-based models, ranking-based methods, and enhanced evaluation metrics. The different methodologies incorporated in this paper contribute to the development of a stronger and more automated Bengali text summarization system, which can be useful in real-world applications like news media, academia, and digital content processing [1].

1.2 Problem Statement

Summarization is a complex task involving understanding the context, finding the salient information, and generating a brief summary. This task is further complicated in Bengali due to scarce large-scale annotated data and linguistic complexities of the language. Although pre-trained transformer models have been found to generate good summaries, it is difficult to select the best summary among the different candidates.

Current summarization models tend to produce summaries that are highly variable in quality, coherence, and informational content. This makes it challenging to identify the "best" summary for a particular piece of text. Furthermore, since summarization is a subjective task, different models can emphasize different parts of the text and thus produce different summaries.

To counter these challenges, the current research proposes a ranking-based approach that evaluates multiple candidate summaries generated by different pre-trained models and selects the most suitable summary based on a given set of evaluation metrics. By combining the strengths of different models, this approach aims to improve the overall quality and accuracy of summarization in the Bengali language.

1.3 Motivation

The motivation driving this thesis stems from the critical need to advance Bangla text summarization systems in the face of linguistic and computational challenges, which can lead to transformative outcomes for both technology and society:

- **Limited-Resource Language Barriers:** Bengali, while having over 250 million speakers, suffers from a lack of large annotated datasets and domain-specific tools to aid in the creation of high-quality summarization models.
- **Model Performance Heterogeneity:** Existing pre-trained models (e.g., mT5 variants) produce variant summaries depending on diverse training data and architectures, necessitating a unified process to select the optimal outputs.
- **Bridging Abstractive-Extractive Gaps:** Abstractive models, though generating new content, may contain factual errors, while extractive models are not creative. A hybrid ranking-based method can reconcile these trade-offs
- **Real-World Applications:** Better summarization systems can transform areas such as news aggregation, educational content curation, and legal document analysis in Bangladesh, facilitating quicker decision-making and resource optimization.
- **Scalability to Low-Resource Environments:** By emphasizing light-weight ranking algorithms rather than retraining large models, this approach ensures scalability to low-resource environments with minimal computational power.

This research addresses these challenges by integrating state-of-the-art pre-trained models and leveraging ranking mechanisms to deliver accurate, context-aware summaries, thereby advancing NLP capabilities for Bangla and similar low-resource languages.

1.4 Objectives

The primary objectives of this research are as follows:

- **Model Selection and Integration:** Identify and integrate multiple pre-trained transformer models for Bengali text summarization to generate diverse candidate summaries.
- **Ranking-Based Summarization:** Develop a ranking-based approach that evaluates and ranks candidate summaries based on their similarity to a reference summary, using metrics such as BLEU, ROUGE, BERTScore, and METEOR.
- **Performance Evaluation:** Evaluate the effectiveness of the proposed approach by comparing the ranked summaries against human-annotated reference summaries using standard **NLG** evaluation metrics.
- **Public Availability:** Make the implementation of the proposed approach publicly available to encourage further research and collaboration in the field of Bangla text summarization.

By fulfilling these objectives, this study aims to contribute to the field of Bangla text summarization and offer a useful contribution for those interested in effectively summarizing information from Bangla texts.

1.5 Organization of the Thesis Document

This thesis's clear message is organized as follows:

Chapter 2 - Literature Review: This chapter summarized related research work and described research work comparison.

Chapter 3 - Overview of Methodology, Transformer Learning Algorithms and Ranking Based Algorithm: This chapter summarized the overview of Transformer Learning Algorithms, Ranking Based Algorithm and methodology and its working method.

Chapter 4 - Required Tools: All of those tools that we used in my work were outlined in this chapter.

Chapter 5 - Performance Evaluation: The results and discussion regarding the system's achievement in multiple Transformer learning models were summarized in this chapter.

Chapter 6 - Conclusion and Future Work: This chapter summarized the result's conclusion. It also includes our limitations and future work for this outcome.

CHAPTER 2

LITERATURE REVIEW

2.1 Related Work

The paper "Rank Your Summaries: Bengali Text Summarization Enhanced through Ranking-based Approach" presents a new method for enhancing Bengali text summarization using various pre-trained transformer models combined with a ranking mechanism. Four various pre-trained summarization models are employed in the paper for generating likely summaries, which are then evaluated and ranked through a graph-based TextRank algorithm. Ultimately, the top-ranked summary is produced as the output. The paper evaluates the approach using standard NLG metrics such as BLEU, ROUGE, BERTScore, WIL, WER, and METEOR through comparisons of the summaries generated against human-annotated references. The results demonstrate that the approach significantly enhances the accuracy and efficiency of Bengali text summarization. The authors also make their implementation publicly available to enable further research in Bengali natural language generation. One limitation of the study is its reliance on pre-trained models, which may not have sufficient generalization capabilities over different types of Bengali texts. [1]

[2] The paper "Bengali News Abstractive Summarization: T5 Transformer and Hybrid Approach" introduces a hybrid model for Bengali news summarization through the combination of BenSumm model for extractive summarization and BanglaT5 model for abstractive summarization. The paper analyzes the performance of the suggested hybrid model and the individual T5 model based on ROUGE and BLEU metrics. The findings demonstrate that the T5 model performs better than the hybrid model with higher ROUGE and BLEU scores. The research identifies the superiority of transformer-based models for creating abstractive summaries of Bengali language texts. A limitation of the research is that it is limited to news articles, which indicates the necessity to conduct summarization on other domains, including medical literature and academic textbooks.

[3] The paper titled "Advancing Abstractive Bangla Text Summarization: A Deep Learning Approach Using Seq2seq Encoder-Decoder Model and T5 Transformer" seeks to respond to the limited number of abstractive Bangla text summarization

techniques through an in-depth analysis based on two deep learning models: the sequence-to-sequence (seq2seq) encoder-decoder model and the T5 transformer. The work compares the models' performance through BERT and ROUGE scores on the AUST NLP Research dataset and the BNLPC datasets created. The findings indicate that the Seq2seq model outperforms the T5 transformer in generating human-like abstractive summaries with an F1-score average of 0.8551. The study emphasizes the necessity of deep learning models to understand intricate language patterns and generate summaries akin to human authors. The study's limitation is that it employs a small dataset size, and there is a need for larger and more varied datasets for better model performance.

[4] The paper "Abstractive Bengali Text Summarization Using Transformer-based Learning" presents an automatic abstractive text summarization system for Bengali language using transformer models. The study compares the performance of five different transformer-based approaches, including Bangla-T5, mT5, and mBART-50, using datasets including XLSum and BANS. The paper demonstrates that the Bangla-T5 model fine-tuned from a shared benchmark dataset outperforms other methods, achieving a highest ROUGE-2 score of 13.83 in the test set. The paper highlights the challenges involved in Bengali text summarization due to a lack of standardized datasets and advanced computational frameworks. The main contributions include the development of an automatic abstractive summarization system, in addition to an exploration of different transformer models for the purpose of finding a state-of-the-art model for Bengali text summarization. A significant limitation of the research is its reliance on available datasets, suggesting the need for more diverse and larger datasets to enable further improvements.

[5] The article "XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages" presents XL-Sum, a dataset consisting of 1 million pairs of articles and summaries in 44 languages, including low-resource languages. It is extracted from BBC News, providing consistency and high abstraction. Fine-tuning mT5 on XL-Sum results in ROUGE-2 scores of over 11, showing efficacy in multilingual and low-resource summarization. The work presents the largest abstractive summarization dataset with uniform summaries, multilingual benchmarks, and publicly available data to drive research. It demonstrates multilingual training benefits, especially in related languages. As valuable as it is, low-resource languages need more data for improvement. The researchers propose XL-Sum as a baseline for future studies in

multilingual and cross-lingual summarization that bridges NLP gaps for low-resource languages.

[6] The paper "CrossSum: Beyond English-Centric Cross-Lingual Summarization for 1,500+ Language Pairs" presents CrossSum, a large-scale cross-lingual summarization dataset with 1.68 million article-summary pairs for 1,500+ language pairs. Different from existing datasets, which are predominantly English-centric, CrossSum facilitates summarization from any source language to any target language without needing English as a bridge. The dataset is built by matching parallel articles from XL-Sum through cross-lingual retrieval and LaBSE embeddings, which guarantees high-quality article-summary pairs. The paper also introduces a Multistage Language Sampling (MLS) training strategy to train multilingual models and proposes LaSE, an embedding-based evaluation metric that has good correlation with ROUGE and even applies to languages without reference summaries. Experiments demonstrate that an m2m summarization model trained on CrossSum performs better than o2m and m2o models. The paper points out the dataset's promise in taking summarization beyond high-resource languages, even though low-resource languages need to be studied further.

[7] The article "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer" presents mT5, a multilingual version of the T5 (Text-to-Text Transfer Transformer), pre-trained on a massive corpus named mC4, spanning 101 languages. Extending the T5's encoder-decoder architecture, mT5 continues with a text-to-text paradigm, with both input and output being sequences of text. The model is pre-trained with a span-masked language modeling objective over Common Crawl data to achieve wide linguistic coverage. The paper benchmarks mT5 on various multilingual NLP benchmarks and achieves state-of-the-art results on tasks such as XNLI, XQuAD, and MLQA. The paper also suggests fine-tuning with unlabeled pre-training data to overcome issues such as accidental translation in zero-shot learning, and this greatly boosts performance. The authors make the pre-trained models and code available, rendering mT5 a significant resource for multilingual NLP tasks, even though there are difficulties in supporting extremely low-resource languages.

[8] The paper "Bengali Text Summarization with Attention-Based Deep Learning" presents a novel abstractive text summarizer for Bengali with an encoder-decoder framework using an attention mechanism. The research utilizes a Long Short-Term

Memory (LSTM) network model under the seq2seq approach to abstractively summarize Bengali text. The paper uses a dataset of news articles, and the performance of the model is measured with ROUGE metrics. The results show that the model generates meaningful summaries, with a ROUGE score of 0.66. The paper emphasizes the challenges of Bengali text summarization due to the lack of high-quality data and the complexity of the language. The study's limitation is the small dataset size, suggesting the need for larger datasets and further model improvements.

[9] The paper "BanglaBERT: Language Model Pretraining and Benchmarks for Low-Resource Language Understanding Evaluation in Bangla" presents BanglaBERT, a BERT-derived Natural Language Understanding (NLU) model pretrained in Bangla. The research gathers 27.5 GB of Bangla pretraining data and presents two downstream task datasets on question answering and natural language inference. The article benchmarks BanglaBERT on four varied NLU tasks, including text classification, sequence labeling, and span prediction. The findings demonstrate that BanglaBERT performs state-of-the-art, surpassing the multilingual and monolingual counterparts. A limitation of the study is the use of web-crawled data, which could contain noisy or inappropriate content, therefore necessitating more effective data filtration systems.

[10]The paper "Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering" presents a novel Paragraph Ranker aimed at boosting the performance of open-domain question answering (QA) systems. The authors describe an approach that ranks the extracted paragraphs from retrieved documents in a systematic way, thereby enhancing answer recall while eliminating irrelevant content simultaneously. The Paragraph Ranker employs a Bi-LSTM network for representing paragraphs and questions and a similarity function to facilitate paragraph ranking. The method is tested on four open-domain QA datasets, with an average gain of 7.8% on exact match scores. The paper emphasizes the need for paragraph ranking in order to enhance the performance of QA systems, particularly in cases when the classical information retrieval systems are unable to retrieve pertinent documents. The study limitation is the absence of customization in the Document Reader, which can be used to further optimize the performance of the QA pipeline

[11]The paper "Sentence Similarity Measurement for Bengali Abstractive Text Summarization" discusses different techniques for sentence similarity measurement

in Bengali abstractive text summarization. The authors present an approach based on Word Mover's Distance (WMD) to measure the similarity between human summaries and machine summaries. The paper also contrasts WMD with other similarity measures such as Cosine Similarity and Jaccard Similarity, and the finding is that WMD provides more accurate results for the text of a Bengali language. The authors use a dataset collected from online and social media, and they generate summaries by utilizing a Bi-directional RNN with LSTM. The paper highlights the importance of measuring sentence similarity for improving the quality of abstractive summarization systems. The study is limited in its scope by concentrating on short Bengali texts, calling for additional research on longer texts.

[12]The paper "TextRank: Bringing Order into Texts" presents the TextRank algorithm, a graph-based ranking approach for natural language processing applications like keyword extraction and sentence extraction for summarization. The authors show how TextRank can be used to construct graphs from text, with vertices being words or sentences, and edges being relations like co-occurrence or similarity. The algorithm orders the vertices by their significance in the graph, and the highest-ranking vertices are chosen to be included in the summary. The paper tests TextRank on both keyword extraction and sentence extraction tasks, and it performs better than state-of-the-art systems in both respects. The authors highlight TextRank's unsupervised characteristic, which allows it to be easily applied to other languages and domains. The study limitation is that TextRank is based on lexical and syntactic information that does not pick up on deeper semantic relations within the text.

[13]The Paper "ROUGE: A Package for Automatic Evaluation of Summaries" presents the ROUGE evaluation package, which was specifically created for automatic summarization quality evaluation through reference to human-annotated reference summaries. The package contains a set of metrics, including ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S, based on n-gram co-occurrence statistics, longest common subsequences, weighted longest common subsequences, and skip-bigram co-occurrence, respectively. The authors test the adequacy of these measures based on analysis data from the Document Understanding Conference (DUC) and demonstrate that ROUGE measures are strongly correlated with human judgments. The paper highlights the need for automated evaluation methods in summarization research, especially when dealing with large-scale evaluation where human evaluation is not feasible. The study limitation is that ROUGE measures are mainly

based on lexical overlap and might not completely reflect the semantic quality of summaries.

[14]The article "BLEU: A Method for Automatic Evaluation of Machine Translation" presents BLEU (Bilingual Evaluation Understudy), an automatic evaluation metric for machine translation quality. BLEU is presented as a solution to the drawbacks of human evaluation, being costly, time-consuming, and not reusable. BLEU calculates the proximity of a machine translation to one or several human reference translations by using n-gram precision with some modifications for brevity penalties. The approach takes into account different n-gram overlaps with the reference translations and the candidate translation, with a bias for higher-order matches to reward fluency. BLEU is shown to be effective through strong correlation with human judgments, and its efficiency in computation makes it viable for large-volume and frequent evaluation. The research points out BLEU's strength in differentiating between adequate and bad translations, even though it might not always reflect meaning fidelity or syntactic accuracy. In spite of some drawbacks, BLEU has been a de facto standard for machine translation evaluation.

[15]The paper "From WER and RIL to MER and WIL: Better Evaluation Measures for Connected Speech Recognition" introduces two novel evaluation measures, the Match Error Rate (MER) and Word Information Lost (WIL), for evaluating the performance of connected speech recognition (CSR) systems. According to the authors, the existing measure, Word Error Rate (WER), is inadequate for applications where the main objective is not error correction but the delivery of information. MER quantifies the ratio of incorrect word matches, and WIL estimates the ratio of word information lost due to recognition. The paper demonstrates that both MER and WIL give more informative performance measures for CSR systems, particularly in high error rate situations. The authors also comment on the best alignment process for computing HSDI (Hits, Substitutions, Deletions, and Insertions) counts, which are then utilized to calculate MER and WIL. The limitation of the study is that the proposed measures are still based on word-level information and may not fully capture the semantic content of the recognized speech.

[16]The paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" presents BERT, a novel language representation model that pre-trains deep bidirectional representations from unlabeled text. The paper shows that BERT

can be fine-tuned with only one extra output layer to generate state-of-the-art models for a diverse set of NLP tasks, such as question answering and language inference. The article emphasizes the significance of bidirectional pre-training and masked language models (MLM) and next sentence prediction (NSP) tasks. The findings indicate that BERT obtains new state-of-the-art results on eleven NLP tasks, surpassing the performance of current models. The limitation of the study is the computational expense of pre-training, proposing the necessity of more efficient training procedures.

[17]The paper titled "BERTSCORE: Evaluating Text Generation with BERT" suggests an automatic text generation evaluation score utilizing contextual embeddings of pre-trained BERT models. In this work, the efficacy of BERTSCORE is benchmarked against that of conventional evaluation metrics, i.e., BLEU and METEOR, on 363 diverse machine translation and image captioning systems. The findings indicate that BERTSCORE has greater correlation with human judgment and gives improved model selection performance compared to the previous approaches. The paper emphasizes the limitations of conventional n-gram-based metrics at capturing paraphrasing and semantic meaning, which is resolved by BERTSCORE using contextual similarity calculation. The key contributions are the proposal of an embedding-based scoring system for the evaluation of text generation and an empirical validation of its efficacy on several tasks. An important limitation of the study is the dependence on pre-trained transformer models, which could be variably available across all languages, decreasing its generalizability.

[18]The paper "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments" introduces an automatic machine translation evaluation metric with better correlation with human judgments that builds on standard n-gram-based approaches. The research compares the performance of METEOR with BLEU and NIST based on unigram recall, stemming, synonym matching, and fragmentation penalties for enhancing the correlation with human evaluation scores. The study illustrates that METEOR achieves a better correlation with human judgments than BLEU, especially at the segment level, by relieving recall problems and word order variation issues. Further, the study emphasizes BLEU's limitations in punishing translations that express equivalent sense using different words. The key contributions are the creation of a flexible evaluation measure that considers semantic similarity and syntactic flexibility issues. A major

shortcoming of the study is that it relies on external linguistic resources and therefore complete functionality is only possible for a subset of languages.

[19]The paper "PageRank Algorithm" presents a link analysis algorithm used for ranking web pages based on hyperlink structures. The study explains the iterative computation of PageRank scores, which assign importance to a page based on the number and quality of incoming links. The paper shows that PageRank is a successful modeling of Web navigation considered as a Markov process and that this ranking is stable even for networks of very large size. This paper identifies the damping factor as critical in controlling the probability of the random jump between pages. Its major contributions include the formalization of a scalable and robust ranking algorithm that finds wide adoption in search engine optimization and information retrieval. One major shortcoming with the research findings is the vulnerability of PageRank to link manipulation, which may be exploited in artificially boosting the rankings.

[20]The paper "Implementing Deep Learning-Based Approaches for Article Summarization in Indian Languages" explores text summarization for low-resource Indian languages using deep learning techniques. The study focuses on summarization of news articles in English, Hindi, and Gujarati using pre-trained sequence-to-sequence (seq2seq) models fine-tuned on the ILSUM 2022 dataset. The research highlights PEGASUS as the best-performing model for English, IndicBART with data augmentation for Hindi, and PEGASUS with a translation-mapping approach for Gujarati. Evaluation is conducted using ROUGE-1, ROUGE-2, and ROUGE-4 metrics, showcasing the effectiveness of these models. The major contribution of this paper is the adaptation and fine-tuning of state-of-the-art models for Indian languages, demonstrating scalable and robust summarization techniques. However, a key limitation is the lack of large, high-quality datasets for low-resource languages, affecting model performance compared to English. The study suggests future improvements in dataset quality, preprocessing, and computational resources to bridge this research gap.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter provides an overview of the research technique, offering insights into preprocessing, system architecture, and concludes by presenting the best model discovered. Figure 3.1 illustrates the proposed approach.

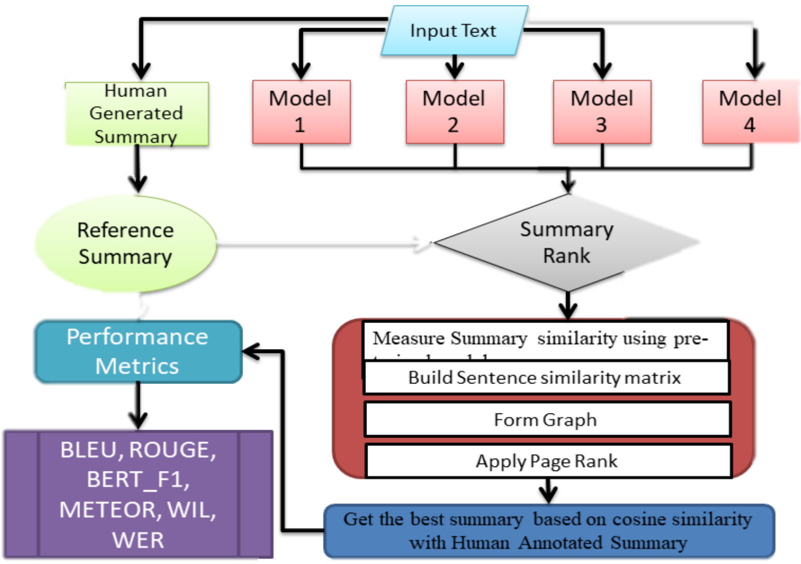

Figure 3.1: Methodology

The proposed methodology, therefore, is a structured ranking-based approach towards the improvement of Bengali text summarization, which selects a summary that is best among all the generated multiple pre-trained model outputs. It starts with the preprocessing of the input Bengali text-eliminating special characters and punctuation from the text. Four pre-trained transformer-based models, namely, mT5 XLSum, mT5 CrossSum, SciBERT Uncased, and mT5 by Shahidul, are used for generating multiple candidate summaries for every input text. In order to obtain the most informative and correct summary, a graph-based ranking technique is utilized called TextRank. First, similarities between the candidate summaries and the human-annotated reference summary are measured by BanglaBERT embeddings and cosine similarity. A summary similarity matrix (SSM) compares the reference summary with each candidate summary and assigns similarity scores by comparison. A summary similarity matrix is then built by comparing the reference summary against each candidate summary, assigning similarity scores according to the comparison. Further, a graph is drawn with the summaries as nodes and the similarity scores as weighted edges, and the summaries are ranked using the PageRank algorithm. The highest-ranked summary is chosen as the final product. The performance of this approach is evaluated by comparing the selected summary against human-annotated summaries based on a number of shared NLG metrics, including BERTScore for semantic similarity, METEOR for unigram matching, WER for word error rate, ROUGE for content overlap, BLEU for n-gram overlap, and WIL for lost information measurement. It has validated the methodology using two benchmark datasets: Bangla Text Summarisation dataset and the Bengali portion of the XL-Sum dataset. Through tests, it is shown that the presented ranking-based method, while integrating the advantages of a number of models, greatly enhances summary selection. These hyperparameters include beam search size set to 4, a minimum of 64 tokens, but a maximum summary length of 400 tokens, and no repeat n-gram size set to 2-all to ensure that the generated summaries are diverse yet coherent. Results indicate that indeed, abstractive models produce more concise and logical summaries, even when the extractive methods are good at preserving word-level overlaps. By weighing all these advantages in a proper manner, the ranking technique generates enhanced summaries with quality outperforming each summarization algorithm.

3.2 Dataset and Data collection

The *Bangla Text Summarization dataset* is a large-scale Bengali-language dataset designed for abstractive and extractive text summarization tasks. It contains **80.3K** data points, each consisting of three attributes:

- **Category** – The topic or domain of the text, which helps in categorizing the content.
- **Text** – The full-length Bengali document or article that needs to be summarized.
- **Summary** – A human-written summary that provides a concise version of the main text.

Although the dataset has **80.3K** records, in this particular study, a subset of **2,000** randomly selected samples was used for experimentation. The dataset, available on Kaggle, is one of the few large-scale Bengali text summarization resources, making it valuable for training and evaluating summarization models. It is suitable for **abstractive and extractive summarization** approaches and can be used to fine-tune transformer-based models like **BanglaBERT, mT5, and BanglaT5**. Few data sample are shown in Figure 3.2 and in Figure 3.3 Graph of Distribution of Text Lengths showed.

	category	summary	text
0	technology	অ্যাপসে মিলবে ঢাকাসহ তিন জেলা আদালতের তথ্য	ঢাকা মহানগর ও ঢাকা জেলা আদালত, কিশোরগঞ্জ ও রাঙা...
1	bangladesh	বিজ্ঞান ও প্রকৌশলে মার্কিন সর্বোচ্চ সম্মাননা...	যুক্তরাষ্ট্রে বিজ্ঞান ও প্রকৌশলে পেশা শুরুর প...
2	bangladesh	বিকল্প শিশুখাদ্য গ্রহণে শিশুর মৃত্যু হলে শাস্ত...	বিকল্প শিশুখাদ্য ও বাণিজ্যিকভাবে উত্পাদিত শিশু...
3	bangladesh	ট্রেনে কাটা পড়ে সাবেক সিভিল সার্জনের মৃত্যু	বগুড়ার আদমদীঘির সান্তাহারে গতকাল শনিবার ট্রেন...
4	bangladesh	যাত্রাবাড়ীতে চুলা জ্বালাতে গিয়ে দুই কর্মচারী ...	যাত্রাবাড়ীর একটি রেস্তোরাঁয় গতকাল বুধবার ভোর...

Figure 3.2: Visualization of Dataset(before pre-processing)

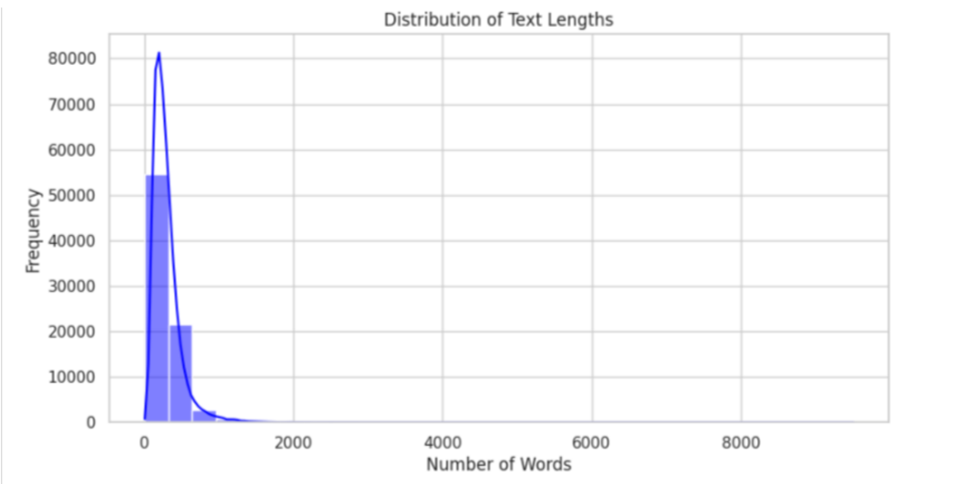

Figure 3.3 : Visualization of Dataset

3.2.1 Attribute Information and Data Summary

Bar Chart for Category Distribution:

The bar chart visually represents the distribution of different categories within the dataset. Using sns.countplot(), the categories are plotted on the **y-axis**, while their respective counts appear on the **x-axis**. The bars are ordered based on frequency, ensuring that the most common categories are displayed at the top. The **'viridis'** color palette enhances readability, making it easy to compare category frequencies at a glance. This visualization helps identify dominant and less frequent categories in the dataset. In figure 3.4 Category distribution visualized.

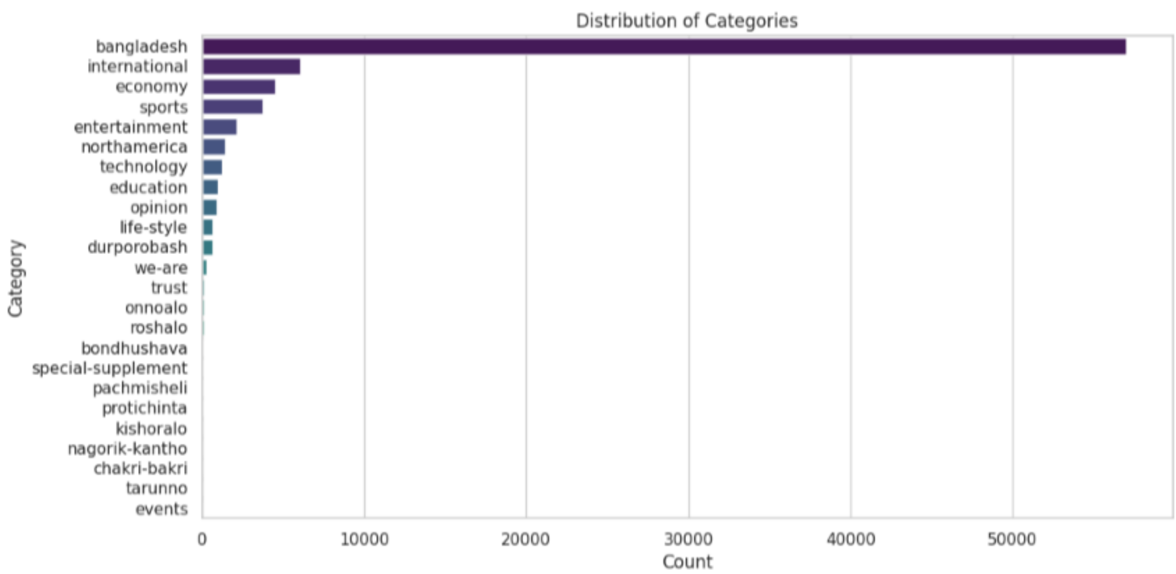

Figure 3.4 : Visualization of Categories

Histogram of Summary & Text Columns:

This code visualizes the distribution of text lengths for both the **Summary** and **Text** columns using histograms. It first calculates the character count for each entry, removes NaN/infinite values, and then plots two side-by-side histograms. The **green** histogram represents **summary lengths**, while the **blue** histogram represents **text lengths**, both with KDE curves for better insights. This helps analyze the variation in text sizes across the dataset. The histogram is visualized in figure 3.5.

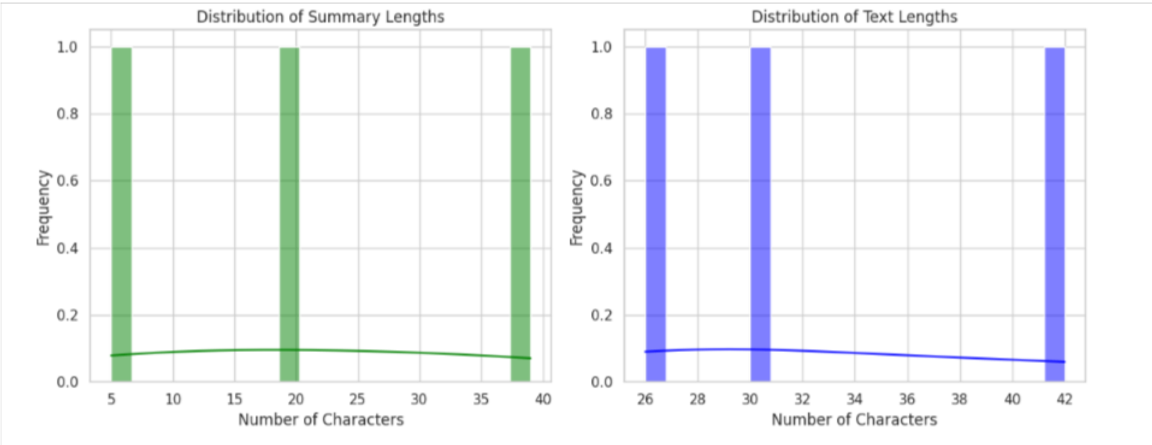

Figure 3.5 : Visualization of Distributed Summary and Text length

Missing Values Heatmap:

The heatmap appears mostly blank because the dataset has very few missing values (only 8 in the **text** column) compared to its total size (**80,331 rows**). This makes missing values difficult to spot visually. To enhance visibility, using a **different color scheme** (e.g., **coolwarm** or **Reds**) can improve contrast. Alternatively, a **bar chart of missing value percentages** provides a clearer representation of missing data distribution across columns. For this reason in figure 3.6 we find nothing because of lowest missing values and that's why in figure 3.7 here I used an alternative bar chat in percentages.

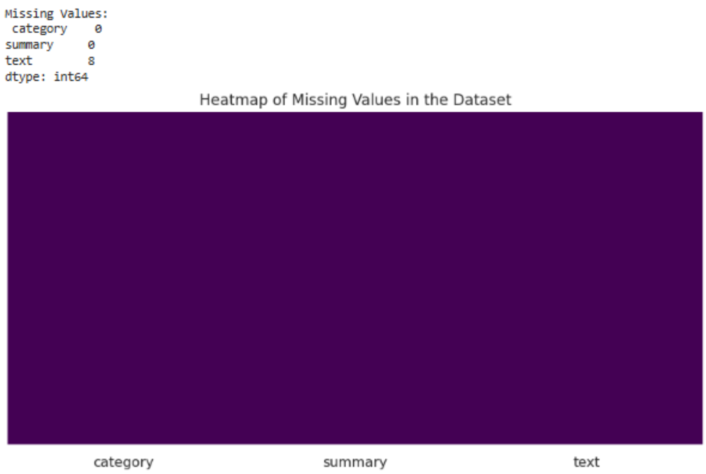

Figure 3.6 : Visualization of Heatmap Missing Values

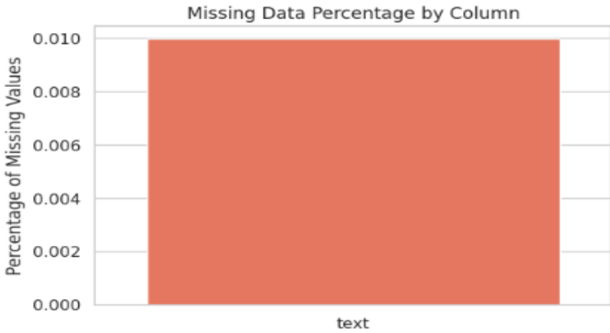

Figure 3.7 : Visualization of Missing Data (Percentage)

Text and Summary Word Cloud:

This code visualizes word clouds for both full Bengali text and its summary using the WordCloud library. A Bengali font is loaded for proper text rendering. The most frequent words appear larger, helping to compare key terms between the text and its summary. The visualization is generated using matplotlib, with a colorful ‘rainbow’ colormap for better readability. This helps analyze how well the summary captures essential themes. Here in figure 3.8 we get the Word Cloud for both text and summary.

Figure 3.7 : Visualization of Word Cloud(Text and Summary)

3.3 Preprocessing

Data preprocessing plays an important role in effectiveness and accuracy in this research work. Several steps related to cleaning, normalization, and preparation of the dataset were performed before feeding into the transformer-based model for Bangla text summarization. This phase was very important to maintain data quality and optimize the performance of the models.

3.3.1 Text Cleaning

Text cleaning is an essential step to remove unwanted elements such as extra spaces, newline characters, and special symbols that might interfere with text processing. Ensuring the text is free from unnecessary formatting helps maintain clarity and consistency. It also prevents unexpected errors when tokenizing the data. By eliminating redundant characters, the input text becomes more structured and suitable for further processing. This step ensures that only meaningful and relevant content is fed into the model. The figure 3.3.1 showed the uses of ‘cleanText’ function.

```
def cleanText(text2clean):
    empty_string = ""

    space = chr(32)
    flag  = 0

    for i in range(len(text2clean)):
        if text2clean[i] != "'" and text2clean[i] != "\n":
            if text2clean[i] == space and flag == 0:
                flag = 1
                empty_string = empty_string + text2clean[i]
            elif text2clean[i] != space:
                empty_string = empty_string + text2clean[i]
                flag = 0
    return empty_string
```

Figure 3.3.1 : Text Cleaning Function

- The ‘cleanText()’ function removes **extra spaces** and **newline characters**.
- It also ensures that multiple spaces do not appear consecutively.

The `cleantext` function in text preprocessing is employed to eliminate unwanted characters, punctuation, numbers, and additional spaces in order to have clean and organized text for NLP. It aids in the standardization of text by making it lowercase and removing noise such as special symbols. Regular expressions (`re.sub()`) are typically utilized to accomplish this cleaning in an effective way. The function enhances text quality for applications such as text summarization, classification, and sentiment analysis. Customization features provide for changes according to particular language and use case requirements.

3.3.2 Whitespace Handling

Whitespace inconsistencies, such as multiple spaces and extra line breaks, can distort text representation. A `WHITESPACE_HANDLER` function is defined using regular expressions (`re.sub()`) to replace redundant spaces and newlines with a single space. This ensures that input text remains uniformly formatted and structured. Proper whitespace handling prevents tokenization errors and enhances text readability. A well-structured text input allows the transformer model to process data more efficiently. Figure 3.3.2 showed the implement of 'whitespace handler' function.

```
WHITESPACE_HANDLER = lambda k: re.sub('\s+', ' ', re.sub('\n+', ' ', k.strip()))
```

Figure 3.3.2 : Whitespace Handling Function

- Uses **regular expressions (re.sub)** to replace multiple spaces and newlines with a single space.

3.3.3 Bangla Text Normalization:

Normalization is a crucial process that ensures the text follows a standardized format, making it easier for the model to interpret. A specialized Bangla text normalizer was used to correct inconsistencies such as spelling variations and unnecessary punctuation. This step helps maintain linguistic accuracy, which is especially important in a morphologically rich language like Bangla. Normalization also improves text alignment with the model's training data, enhancing summarization performance. Ultimately, it ensures that input text remains clean, readable, and well-structured. This normalizer importing has been showed in the figure 3.3.3.

```
import re
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM
from normalizer import normalize # pip install git+https://github.com/csebuetnlp/normalizer
import torch
```

Figure 3.3.3: Implementation of Nomalizer

3.3.4 Loading the Pretrained Model and Tokenizer:

All the pre-trained models and tokenizer are loaded using the `transformers` library. The tokenizer converts raw text into numerical tokens, while the model generates summaries. Using a pretrained transformer model significantly enhances the summarization quality. The tokenizer ensures that the input text is processed correctly for the model. Loading the correct model and tokenizer is crucial for accurate and efficient text summarization.

3.3.5 Setting Up Language ID Mapping:

The script retrieves the language ID for Bengali from the model's configuration. This is necessary for correctly encoding the text before generating summaries. Setting the correct language ID ensures that the model recognizes and processes the input text accurately. This mapping helps the model differentiate between multiple languages and optimize token generation. Without this step, the summarization output may not align with the intended language. The application of the 'get_lang_id' function showed in figure 3.3.5.

```
get_lang_id = lambda lang: tokenizer._convert_token_to_id(
    model.config.task_specific_params["langid_map"][lang][1]
)

target_lang = "bengali"
```

Figure 3.3.5: Implementation of Language ID Mapping

3.3.6 Tokenization & Numerical Encoding:

Before feeding text into a transformer model, it must be converted into a numerical format that the model can understand. Tokenization breaks the text into smaller units, such as words or subwords, and maps them to corresponding numerical representations. This step enables the model to process text efficiently and generate meaningful summaries. Proper tokenization ensures that even complex words are handled effectively, improving overall summarization accuracy. By encoding the text correctly, the model can better recognize language patterns and relationships.The maximum token length was set to 512 to prevent excessive truncation while ensuring efficient processing.

- **Tokenizes the input text** using tokenizer(article_text, return_tensors="pt", padding="max_length", max_length=512, truncation=True).input_ids

3.3.7 Moving the Model to GPU (if available):

GPU availability using `torch.cuda.is_available()` and moves the model to `cuda` if a GPU is present; otherwise, it defaults to CPU. This ensures faster computation and efficient processing of large datasets. Running the model on GPU significantly reduces inference time. Utilizing hardware acceleration optimizes model performance and allows for large-scale text summarization. This step ensures that processing remains efficient, even for complex and lengthy inputs.

3.3.8 Creating an Empty DataFrame for Storing Results:

A `pandas` DataFrame is initialized to store the original text, given summaries, and the model-generated summaries. This structured approach helps in evaluating and analyzing the summarization quality. Storing results in a structured format allows for easier visualization and assessment. The DataFrame serves as an organized repository for tracking summarization performance. This ensures that all generated outputs are systematically recorded for further analysis.

3.3.9 Processing and Summarizing the Dataset:

This step ensures that all inputs undergo consistent processing for optimal summarization results after completing the all pre-processed steps mentioned above. At the ending of pre-processing, the pre-processed summary will be generated which is showed in figure 3.3.9.

0	বাংলাদেশের সব আদালতের মামলার সংক্রান্ত তথ্য এখন থেকে মোবাইল অ্যাপসে পাওয়া যাবে বলে জানিয়েছে দেশটির প্রধান বিচারপতি সুরেন্দ্র কুমার সিনহা। আজ শনিবার ঢাকা মহানগর দায়রা জজ আদালত, কিশোরগঞ্জ এবং রাঙামাটি জেলার বেশ কয়েক
1	যুক্তরাষ্ট্রে বিজ্ঞান ও প্রকৌশলে পেশা শুরুর প্রাথমিক পর্যায়ের গবেষকদের সর্বোচ্চ সম্মাননা হলো 'প্রেসিডেনশিয়াল আরলি ক্যারিয়ার অ্যাওয়ার্ড ফর সায়েন্টিস্টস অ্যান্ড ইঞ্জিনিয়ার্স'।
2	বাংলাদেশের স্বাস্থ্য মন্ত্রণালয়-সম্পর্কিত সংসদীয় কমিটি 'মাতৃদুগ্ধ বিকল্প শিশুখাদ্য ও বাণিজ্যিকভাবে উত্পাদিতশিশু খাদ্য' বিলে দণ্ডবিধির আওতায় শাস্তিমূলক ব্যবস্থা নেওয়ার সুপারিশ করেছে।
3	বাংলাদেশের বগুড়ার আদমদীঘির সান্তাহারে গতকাল শনিবার ট্রেনের নিচে কাটা পড়ে একজন সাবেক সিভিল সার্জন মারা গেছেন বলে পুলিশ জানিয়েছে। এই খবর নিশ্চিত করেছে নওগাঁর শিবগঞ্জ উপজেলার বাঙ্গাবাড়িয়ার পরিবারের সদস্যরা।
4	বাংলাদেশের রাজধানী ঢাকার যাত্রাবাড়ি মোড় এলাকায় কেরোসিনের চুলা জ্বালানোর সময় দুই কর্মচারী অগ্নিদগ্ধ হওয়ার খবর পাওয়া যাচ্ছে। কিন্তু তাঁদের চিকিৎসা দেওয়া হয়েছে হাসপাতালে।
5	বাংলাদেশের রাজধানী সিলেটসহ একাধিক জেলা বা উপজেলাকে বাল্যবিবাহমুক্ত ঘোষণা করা হয়েছে। কিন্তু সরকারি কর্মকর্তারা এ বিষয়ে তথ্য-উপাত্ত হাতে নিয়েই কোনো জেলার বাণিজ্যিক ব্যবস্থা গ্রহণ করেছে।
6	বাংলাদেশ প্রিমিয়ার লিগ (বিপিএল) আগামীকাল থেকে শুরু হওয়ার কয়েকদিন আগে পাকিস্তানের ক্রিকেট বোর্ডের সঙ্গে অনাপত্তি পত্র (এনওসি) দেওয়ার ক্ষেত্রে একটি নতুন নীতি সৃষ্টি করেছে।
7	লক্ষ্মীপুরের পোদ্দার বাজারে সন্ত্রাসী হামলার শিকার নোয়াখালীর চাটখিল উপজেলা ভাইস চেয়ারম্যানকে গতকাল শুক্রবার ভোরে আহত অবস্থায় উদ্ধার করা হয়েছে বলে জানিয়েছে সরকার।
8	বাংলাদেশের বরিশাল জেলা প্রশাসনের নির্বাহী হাকিম মো. আল-মুক্তাদির হোসেন ও পাপিয়া সুলতানাসহ স্থানীয় ব্যবসায়ীরা একজোট হয়ে দোকানপাট বন্ধ রেখে ওই ব্যবসাপ্রতিষ্ঠানকে জরিমানা করেন।
9	বাংলাদেশে বিএনপির চেয়ারপারসন খালেদা জিয়া নাইকো দুর্নীতি মামলার কার্যক্রমে স্থগিতাদেশ দিয়েছে সুপ্রিম কোর্টের একটি দ্বৈত বেঞ্চ। এই রায়ের বিরুদ্ধে আপিল করেছেন সাবেক প্রধানমন্ত্রী।
10	চাঁপাইনবাবগঞ্জ সদর উপজেলা পরিষদের চেয়ারম্যান সাময়িক বরখাস্ত হওয়ার পর তাঁর দপ্তরের সব উন্নয়নকাজ বন্ধ রয়েছে। এই অবস্থায় নতুন করে তিন কোটি টাকা ফেরত যাওয়ার আশঙ্কা তৈরি হয়েছে।

Figure 3.3.9: Pre-processed Summary Data

3.3.10 Saving the Generated Summaries:

After processing, the generated summaries are saved to a CSV file using `pandas.to_csv()`. This allows easy retrieval and further analysis of the model's output. Saving results ensures that the summarization data remains accessible for future research and evaluation. This step provides a structured way to review and validate the model's performance. Storing the results in a file enables external analysis and comparison with other summarization techniques. The pre-processed dataset need to save so that for each pre-trained models we can get the dataset and marge the all generated summaries. Here for "mT5_m2m_CrossSum_enhanced' pre-trained model generated summary saved to kaggle working which is showed in figure 3.3.10.

```
genDf.to_csv('/kaggle/working/mT5_m2m_CrossSum_enhanced.csv', index=False)
print("Saved the generated summaries to mT5_m2m_CrossSum_enhanced_results.csv")
```

```
Saved the generated summaries to mT5_m2m_CrossSum_enhanced_results.csv
```

Figure 3.3.10: Pre-processed Data Saving

3.4 Bangla sentence similarity

The function bn_sentence_similarity(sent1, sent2) is designed to compute the similarity between two Bangla sentences using a transformer-based model. This is crucial for identifying key sentences in a text summarization task. The process begins with **tokenization**, where both input sentences are converted into numerical representations using the transformer's tokenizer.encode_plus() function. This method encodes the text into token IDs, ensuring a **maximum length of 400 tokens** while truncating longer sentences and padding shorter ones to maintain uniformity. Additionally, an **attention mask** is generated to distinguish between actual words and padded tokens, which helps the model focus only on relevant content. These processed tokenized representations are stored in a dictionary, with 'input_ids' containing the tokenized sentence and 'attention_mask' ensuring that only meaningful parts of the sentence are considered during further computations. The function stacks the tokenized inputs into tensors, preparing them for batch processing.

After tokenization, the function **processes the tokens through a pretrained transformer model** to generate sentence embeddings. These embeddings are obtained from the **last hidden state** of the model's output, which contains dense vector representations of the sentences. These embeddings capture contextual and semantic meaning, making them highly effective for similarity computations. However, since padded tokens should not influence the similarity computation, a **masking technique** is applied. The attention mask is expanded to match the shape of the embeddings, and a multiplication operation ensures that only valid token representations contribute to the final sentence embedding. To further refine the embeddings, **mean pooling** is applied. This step aggregates token embeddings by summing them along a specific axis and then dividing by the sum of valid tokens in the attention mask. As a result, each sentence is represented as a **single dense vector**, making it suitable for similarity comparison.

Once the embeddings are computed, the next step is **calculating cosine similarity** between the two sentence vectors. Before doing so, the function detaches the tensors from the computational graph and converts them into NumPy arrays for efficient numerical processing. The cosine_similarity() function is then used to compute the similarity score between the two sentences. A **cosine similarity score of 1** indicates that the sentences are identical in meaning, while **a score of 0** suggests no similarity. If the score is negative, it implies that the sentences have opposite meanings. This similarity metric provides a numerical measure of how closely two sentences are related in terms of their semantic content.

Measures the cosine of the angle between two sentence vectors.

$$\mathrm{Sim}(A, B) = \cos\theta = \frac{A \cdot B}{\|A\|\,\|B\|}$$

where:

A and B are vectorized representations of two sentences.

This sentence similarity function plays a crucial role in **ranking sentences for text summarization**. It is integrated into the bn_build_similarity_matrix(sentences) function, where a **similarity matrix** is created by comparing each sentence in the input text with the first sentence. These similarity scores form a matrix structure that helps identify the most important and contextually relevant sentences. This similarity matrix is then used in the bn_generate_summary() function, where **PageRank** is applied to rank sentences based on their significance. The highest-ranked sentences are selected to form the final summary. By leveraging sentence embeddings and cosine similarity, this approach enables the extraction of meaningful summaries while preserving the key ideas of the original text.

3.5 Transformer Learning

Transformer learning is a deep learning approach used for natural language processing (NLP) tasks. It is based on the Transformer architecture, introduced by Vaswani et al. (2017) in "Attention is All You Need".

The Transformer model eliminates the need for recurrent networks (RNNs or LSTMs) and relies entirely on a self-attention mechanism, making it faster, scalable, and efficient in handling long-range dependencies in text.

Types of Transformer Learning

There are **three main types** of transformer-based learning models:

A. Encoder-Only Models (BERT, XLM-R)

- These models use only the **encoder** of the transformer.
- They are trained using **masked language modeling (MLM)**, where some words in the input are masked, and the model learns to predict them.
- Used for **classification, named entity recognition (NER), and sentence pair tasks**.

Examples:

- **BERT** (Bidirectional Encoder Representations from Transformers)
- **XLM-R** (Cross-lingual Model – RoBERTa-based)
- **mBERT** (Multilingual BERT)

B. Decoder-Only Models (GPT, GPT-3)

- These models use only the **decoder** of the transformer.
- They are trained using **causal language modeling (CLM)**, where the model predicts the next word in a sequence.
- Used for **text generation tasks** like chatbot responses and story generation.

Examples:

- **GPT-3** (Generative Pre-trained Transformer 3)
- **DialoGPT** (Conversational AI model based on GPT)

C. Encoder-Decoder Models (T5, mT5, BART, mBART)

- These models use **both encoder and decoder**.
- Trained with a **text-to-text format**, meaning **all tasks are converted into text generation** problems.
- Used for **machine translation, text summarization, and question answering**.

Examples:

- T5 (Text-to-Text Transfer Transformer)
- mT5 (Multilingual T5)
- BART (Bidirectional AutoRegressive Transformers)
- mBART (Multilingual BART)

3.5.1 Bengali Summarizer mT5:

Bengali Summarizer mT5, created by tashfiq61, is a pre-trained sequence-to-sequence model tailored for the Bengali text summarization task. It is derived from mT5 (Multilingual T5), an advanced transformer model fine-tuned with a wide range of multilingual datasets. It has been fine-tuned to produce short and informative summaries of Bengali text, making it very helpful for tasks such as news summarization, document summarization, and content creation in Bengali. With the use of functions of the mT5 model, there is guaranteed high-quality summarization while keeping the linguistic nuance unique to the Bengali language. This model is most useful to developers, researchers, and organizations looking to include automated summarization of Bengali text in their applications

3.5.2 mT5_m2m_crossSum_enhanced:

The mT5_m2m_crossSum_enhanced developed by csebuetnlp is a multilingual, sequence-to-sequence pre-trained model that is specifically developed for the task of cross-lingual summarization. It is an extension of mT5, which is a variant of Google's T5 model trained on a massive corpus of multilingual text. The enhanced variant has been thoroughly fine-tuned to carry out tasks related to summarization and is able to generate summaries in various languages, including languages other than the source language. The model was trained using CrossSum, a large-scale dataset containing 1.68 million article and summary pairs across 1,500 language pairs, in this way improving its ability in many-to-many summarization tasks. Its ability to summarize across languages makes it especially useful for a number of applications such as multilingual content creation, news aggregation, and improving accessibility by providing summaries in the reader's desired language

3.5.3 mT5_multilingual_XLSum:

The mT5_multilingual_XLSum model, developed by csebuetnlp, is a multilingual abstractive text summarization model fine-tuned on a sequence-to-sequence task. It is derived from mT5 (Multilingual T5) and has been fine-tuned on the XLSum dataset, which consists of high-quality article-summary pairs for 45 languages. The model can generate concise and readable summaries of text in numerous languages and is extremely useful for applications like news summarization, document summarization, and cross-lingual content creation. With its capacity to process different linguistic structures and subtleties, the model is an extremely useful tool for researchers, developers, and organizations intending to utilize automated multilingual summarization for their applications

3.5.4 Bangla Text Summarization Model:

The Bangla Text Summarization Model, created by shahidul034, is a pre-trained sequence-to-sequence Bengali text summarization model. The model, based on the mT5 (Multilingual T5) model, has been fine-tuned for Bengali text summarization and can generate short and meaningful summaries while preserving the essence of the original text. It is particularly useful for applications such as news summarization, article condensation, and automatic content generation in Bengali. Leveraging the power of transformer-based learning, the model generates high-quality summarization with improved coherence and fluency. It renders the model a useful tool for researchers, developers, and organizations looking to incorporate Bengali-language summarization into their applications

3.5.5 BanglaBERT

The BanglaBERT Generator of csebuetnlp is a pre-trained sequence-to-sequence Bengali natural language generation (NLG) model. It is derived from the BERT model, which has been fine-tuned to produce high-quality Bengali text for a range of applications including text completion, paraphrasing, and content creation. As opposed to the majority of BERT models, which find the most use in encoding applications, this model is fine-tuned for generation applications and thus is highly appropriate for use in automated content generation, chatbots, and text augmentation applications in the Bengali language. By generating fluent and contextually appropriate Bengali text, this model is a great resource for researchers and developers of Bengali NLP applications

3.6 Ranking Based Algorithm

Ranking algorithms are a fundamental module of information retrieval, web search engines, and natural language processing (NLP). Ranking algorithms order content through the assignment of importance scores based on some rules so that the most relevant information can be conveniently retrieved. Ranking algorithms' concept is to sort a list of items by relevance, authority, or importance, and thus are an essential constituent in applications such as web search, document retrieval, and recommender systems.

A ranking-based algorithm generally follows these steps:

- **Data Representation:** The input data is structured into a network, a list, or a matrix. For example, in web search, web pages are treated as nodes in a **graph-based structure**.
- **Feature Extraction:** The algorithm extracts relevant features, such as **link relationships, textual similarity, or click-through rates**, depending on the domain.

- **Scoring Mechanism:** Each item is assigned a score based on predefined rules, **probabilistic models, machine learning techniques, or recursive ranking functions**.
- **Sorting and Selection:** Items are sorted according to their scores, and the **highest-ranked elements** are selected as the final output.

Types of Ranking-Based Algorithms

1. Link Analysis-Based Ranking (PageRank, HITS)

These algorithms are widely used in **web search engines** to determine the importance of web pages based on their link structures.

- **PageRank:** Computes a web page's authority based on the **quality and quantity** of incoming links. It uses a **damping factor** to simulate random jumps between pages, ensuring fairness in rankings.
- **HITS (Hyperlink-Induced Topic Search):** Assigns two scores to each page: **authority (how many quality links point to it)** and **hub score (how well it links to other important pages)**.

2. Text-Based Ranking (TextRank, TF-IDF)

Used primarily in **NLP applications**, these algorithms rank **words, phrases, or sentences** based on their occurrence and contextual relationships.

- **TextRank:** A graph-based algorithm that **identifies important words or sentences** in a document by analyzing how frequently they co-occur with others.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** Measures word importance based on its **frequency in a document relative to its rarity in a corpus**. It is widely used in **document retrieval and text mining**.

3. Learning to Rank (LTR)

A **machine learning-based** approach where an algorithm learns from **human-labeled ranking data** to improve relevance scores.

- **Pointwise LTR:** Treats ranking as a **regression problem**, predicting the relevance of each item individually.

- **Pairwise LTR:** Compares **pairs of items** to determine which one is more relevant.
- **Listwise LTR:** Evaluates the **entire ranked list** and optimizes it for better user satisfaction.

3.6.1 TextRank

TextRank is a graph-based ranking model applied to natural language processing (NLP) problems like keyword extraction and text summarization. TextRank, developed by Rada Mihalcea and Paul Tarau, treats words, phrases, or sentences as vertices of a graph and links them according to linguistic relationships. Analogous to PageRank, it allocates a value for importance to each vertex and updates the scores in iterations according to the topological construction of the relationships.

In TextRank keyword extraction, we build the co-occurrence graph where words that appear together frequently in a text window are linked. The algorithm ranks words based on their connectivity and salience and returns the most important keywords. This is an effective method of determining key terms without any labeled training data and thus is an unsupervised learning technique.

In text summarization, TextRank constructs a graph with sentences as nodes and semantic similarity as edges. The most pertinent sentences that are at the core of the document meaning are assigned higher scores and picked for the summary. Long texts are effectively condensed into brief summaries without sacrificing key information using this approach

3.6.2 PageRank

PageRank, which was invented at Google by Larry Page and Sergey Brin, is a link analysis algorithm that assigns a measure of importance to web pages according to the number and qualities of links they receive. The algorithm represents the web as a directed graph consisting of web pages as nodes and hyperlinks as edges. The page ranks higher if it has many other pages linked to it, particularly if the pages that link to it are themselves high-ranking pages.

Mathematically, the PageRank value of a page is calculated using a recursive algorithm where each page distributes its rank to the pages it links. The formula has a damping factor, normally 0.85, so the possibility of a user randomly jumping to any web page rather than following links can be taken into account. The iterative process of PageRank guarantees that rank values converge over several iterations into a steady-state distribution that precisely captures the relative importance of pages.

One of the strongest points of PageRank is that it is able to resist manipulation over some of the simpler ranking methods based on keyword frequency. Because it relies on the web structure and not just the content, it provides a very good estimate of authority. But it is computationally expensive and requires large-scale matrix operations and thus becomes hard to compute in real-time for very large networks.

3.6.3 Best Summary Model

The best summary model in the research employs a ranking-based transformer approach to enhance Bangla text summarization by combining the outputs of multiple pre-trained models. Instead of relying on a single summarization model, this method generates multiple candidate summaries using four pre-trained models: bengali-summarizer-mt5, mT5_m2m_crossSum_enhanced,mT5_multilingual_XLSum,and Bangla_text_summarization. A graph-based TextRank algorithm is then applied to rank these summaries based on their similarity to human-annotated reference summaries. The model with the highest ranking **score** is selected as the final summary. The evaluation using BLEU, ROUGE, BERTScore, METEOR, WIL, and WER confirmed that the ranking-based approach consistently outperformed individual models by producing summaries with better content retention, fluency, and semantic alignment. This approach significantly enhances the accuracy and coherence of Bangla text summarization, making it more effective for low-resource languages. In the figure 3.6.3 we can see the which pre-train models generate how much best summaries and in figure 3.6.4 we see the marge of all pre-trained model summary and generated best summary.

```
BestSummaryModel
Bengali_summarizer_mt5        404
mT5_multilingual_XLSum        768
Bangla_text_summarization     130
mT5_m2m_crossSum_enhanced     698
Name: BestSummary, dtype: int64
```

Figure 3.6.3: Number of Best Summary Models

	text	givenSummary	Bengali_summarizer_mt5	mT5_m2m_crossSum_enhanced	mT5_multilingual_XLSum	Bangla_text_summarization	BestSummary	BestSummaryModel
0	ঢাকা মহানগর ও ঢাকা জেলা আদালত, কিশোরগঞ্জ ও রাঙ...	আপসে মিলবে ঢাকাসহ তিন জেলা আদালতের তথ্য	বাংলাদেশের সব আদালতের মামলা সংক্রান্ত তথ্য এখন...	বাংলাদেশের সব আদালতের মামলার সংক্রান্ত তথ্য এ...	বাংলাদেশের সব আদালতের মামলার সংক্রান্ত তথ্য এখ...	এনজিও শার্প'র আয়োজনে হাডুডু প্রতিযোগিতার উদ্...	বাংলাদেশের সব আদালতের মামলা সংক্রান্ত তথ্য এখন...	Bengali_summarizer_mt5
1	যুক্তরাষ্ট্রে বিজ্ঞান ও প্রকৌশলে পেশা শুরুর প...	বিজ্ঞান ও প্রকৌশলে মার্কিন সর্বোচ্চ সম্মাননা...	বাংলাদেশে পেশা শুরুর প্রাথমিক পর্যায়ে বিজ্ঞান...	যুক্তরাষ্ট্রে বিজ্ঞান ও প্রকৌশলে পেশা শুরুর প...	যুক্তরাষ্ট্রে বিজ্ঞান ও প্রকৌশলে পেশা শুরুর প্...	রোহিঙ্গাদের শান্তিপূর্ণ প্রত্যাবর্তনে বাংলাদেশ...	বাংলাদেশে পেশা শুরুর প্রাথমিক পর্যায়ে বিজ্ঞান...	Bengali_summarizer_mt5
2	বিকল্প শিশুখাদ্য ও বাণিজ্যিকভাবে উৎপাদিত শিশু...	বিকল্প শিশুখাদ্য গ্রহণে শিশুর মৃত্যু হলে শাস্ত...	বাংলাদেশের জাতীয় সংসদ ভবনে অনুষ্ঠিত কমিটি সংস...	বাংলাদেশের স্বাস্থ্য মন্ত্রণালয়-সম্পর্কিত সং...	বাংলাদেশে বিকল্প শিশুখাদ্য ও বাণিজ্যিকভাবে উৎ...	গণস্বাস্থ্য কেন্দ্রে 'মা, শিশু স্বাস্থ্য ও প্র...	বাংলাদেশে বিকল্প শিশুখাদ্য ও বাণিজ্যিকভাবে উৎ...	mT5_multilingual_XLSum
3	বগুড়ার আদমদীঘির সান্তাহারে গতকাল শনিবার ট্রেন...	ট্রেনে কাটা পড়ে সাবেক সিভিল সার্জনের মৃত্যু	বাংলাদেশে খুলনাগামী তেলবাহী একটি ট্রেন সান্তাহ...	বাংলাদেশের বগুড়ার আদমদীঘির সান্তাহারে গতকাল ...	বগুড়ার আদমদীঘির সান্তাহারে গতকাল শনিবার ট্রেন...	বগুড়ায় ট্রেনে কাটা পড়ে সাবেক সিভিল সঙ্গী কু...	বাংলাদেশে খুলনাগামী তেলবাহী একটি ট্রেন সান্তাহ...	Bengali_summarizer_mt5
4	যাত্রাবাড়ীর একটি রেস্তোরাঁয় গতকাল বুধবার ভোর...	যাত্রাবাড়ীতে চুলা জ্বালাতে গিয়ে দুই কর্মচারী ...	বাংলাদেশে যাত্রাবাড়ী মোড় এলাকায় কেরোসিনের চ...	বাংলাদেশের রাজধানী ঢাকার যাত্রাবাড়ি মোড় এলা...	বাংলাদেশের রাজধানী ঢাকার যাত্রাবাড়ী মোড় এলাক...	সহকর্মীর এক ঘুষিতেই শ্রমিকের মৃত্যু, দুই দফা ক...	বাংলাদেশে যাত্রাবাড়ী মোড় এলাকায় কেরোসিনের চ...	Bengali_summarizer_mt5
...	...	...	...	...	...	...	...	...
1995	সেবার মান বাড়ানোর মাধ্যমে ব্যাংকের ব্যবসা সম্...	পূবালী ব্যাংকের রাজশাহী অঞ্চলের প্রথম শাখা রা...	বাংলাদেশে ব্যাংকের রাজশাহী অঞ্চলের শাখাগুলোর ব...	বাংলাদেশের পূবালী ব্যাংক লিমিটেডের রাজশাহী অঞ্চ...	বাংলাদেশে পূবালী ব্যাংক লিমিটেডের রাজশাহী অঞ্চ...	মানিলন্ডারিং ও জঙ্গি ইউনিট নিয়ে পৃথক অভিযানে ৬...	বাংলাদেশে পূবালী ব্যাংক লিমিটেডের রাজশাহী অঞ্চ...	mT5_multilingual_XLSum
1996	চট্টগ্রাম মেডিকেল কলেজের (চমেক) মর্গে লাশের ময়...	ময়নাতদন্তে অনিয়মের সঙ্গে জড়িতদের চিহ্নিত কর...	বাংলাদেশে চট্টগ্রাম মেডিকেল কলেজের (চমেক) মর্গ...	চট্টগ্রাম মেডিকেল কলেজের (চমেক) মর্গে লাশের ম...	চট্টগ্রাম মেডিকেল কলেজের (চমেক) মর্গে লাশের ময়...	চট্টগ্রামে ইউপি সদস্য ও দালাল, কাজী ও প্রবাসীক...	চট্টগ্রাম মেডিকেল কলেজের (চমেক) মর্গে লাশের ময়...	mT5_multilingual_XLSum
1997	বিচার বিভাগে যত রাজনৈতিক হস্তক্ষেপ না থাকবে, ত...	বিচার বিভাগে রাজনৈতিক হস্তক্ষেপ না থাকাই মঙ্গল...	বাংলাদেশের সুপ্রিম কোর্ট মিলনায়তনে এক অনুষ্ঠা...	বাংলাদেশের সুপ্রিম কোর্টের প্রধান বিচারপতি সু...	বাংলাদেশে বিচারক নিয়োগের ব্যাপারে সরকারের একটি...	'সমুদ্রসীমার সার্বভৌমত্ব সমুন্নত রাখতে নৌবাহিন...	বাংলাদেশে বিচারক নিয়োগের ব্যাপারে সরকারের একটি...	mT5_multilingual_XLSum
1998	প্রধানমন্ত্রী শেখ হাসিনার ছেলে ও তাঁর তথ্যপ্রয...	শফিক রেহমানের জামিনের পরবর্তী শুনানি ১ জুন	বাংলাদেশে প্রধানমন্ত্রী শেখ হাসিনার ছেলে ও তাঁ...	বাংলাদেশে প্রধানমন্ত্রী শেখ হাসিনার ছেলে ও তা...	বাংলাদেশে প্রধানমন্ত্রী শেখ হাসিনার ছেলে ও তাঁ...	সাংবাদিক জামিনে মুক্ত আসামিকে আটকের পর আদালতে ...	বাংলাদেশে প্রধানমন্ত্রী শেখ হাসিনার ছেলে ও তা...	mT5_m2m_crossSum_enhanced
1999	যেসব কাজে কম দক্ষতা প্রয়োজন, এমন কাজের পেশাদার...	অস্ট্রেলিয়ায় স্বল্প দক্ষ কর্মীদের স্থায়ী ভিসার...	বাংলাদেশের সাবক্লাস ৪৮২ কর্মভিসার মধ্যে নতুন ভ...	নতুন বছরের শুরুতেই সুখবর আসছে অভিবাসন খাতে দ...	যেসব কাজে কম দক্ষতা প্রয়োজন, এমন কাজের পেশাদা...	রাশিফলে জেনে নিন কেমন যাবে রোগীর দালালকে গণপে...	বাংলাদেশের সাবক্লাস ৪৮২ কর্মভিসার মধ্যে নতুন ভ...	Bengali_summarizer_mt5

Figure 3.6.4: Generated of Best Summary Models

CHAPTER 4

REQUIRED TOOLS

4.1 Kaggle

Kaggle has been utilized in this study as one of the primary computational and discovery environments. Kaggle is a cloud-based data science and machine learning environment offering access to large datasets, computational resources, and collaboration tools. It is an interactive, community-driven platform on which researchers and data scientists can try out different algorithms, join competitions, and exchange knowledge. Surprisingly, Kaggle has full support for the Python programming language and is fully compatible with some of the most popular libraries such as TensorFlow, Scikit-learn, and Pandas. Kaggle offers free cloud-based GPUs and TPUs, making model training and testing extremely fast without requiring high-end local machines. Kaggle also offers pre-configured Jupyter notebooks, so there is no requirement for long setups and installations. Its ease of use and social elements make Kaggle an ideal place for data-intensive research, testing, and information exchange, enhancing machine learning process productivity significantly

4.2 Python

Python: Python is a dynamically interpreted, high-level programming language known for its emphasis on code readability through the use of significant indentation. Python is widely regarded as an excellent choice for machine learning tasks, which is why we chose it as the primary programming language for implementing our models.

Pandas: Pandas is a Python library that we utilized for data manipulation and analysis in our research.

NumPy: NumPy is another Python library that proved instrumental in our work. It provides robust support for high-performance multidimensional arrays and essential tools for array computation and manipulation.

Scikit-Learn (Sklearn): Scikit-Learn, often referred to as Sklearn, stands as one of the most powerful and widely used machine learning libraries in Python. It offers a comprehensive set of efficient tools for various machine learning and statistical modeling tasks, including classification, regression, clustering, and dimensionality reduction. Sklearn is built upon Python, with a foundation built on NumPy, SciPy, and Matplotlib.

Keras: Keras is a Python-based deep learning API designed to work seamlessly with TensorFlow, a popular machine learning platform. Keras was instrumental in our research, enabling quick experimentation and a rapid transition from concept to results.

Transformers: The Transformers library, developed by Hugging Face, played a key role in implementing state-of-the-art natural language processing (NLP) models in our research. It provides access to a variety of pre-trained deep learning models such as BERT, GPT, RoBERTa, and T5, enabling efficient fine-tuning for tasks like text classification, translation, and summarization. The library includes built-in tokenization and model deployment tools, streamlining the NLP pipeline. It serves as a powerful, open-source solution for integrating transformer-based architectures into machine learning applications.

Sentence Transformers: Sentence Transformers, a specialized framework for generating dense sentence embeddings, proved essential in computing semantic similarity between textual data in our research. This library, built on models like BERT and RoBERTa, efficiently processes text into vector representations, making it ideal for document clustering, information retrieval, and text ranking. It offers a simple interface for training and applying sentence embeddings, ensuring accurate and high-performance NLP applications

NLTK (Natural Language Toolkit): NLTK, a comprehensive Python library for linguistic processing, played a key role in our research for text tokenization, stemming, lemmatization, and parsing. It provides various tools for natural language understanding (NLU), enabling the efficient extraction of key linguistic features. NLTK's extensive dataset and pre-built models enhance text-processing pipelines, making it a versatile open-source library for syntactic and semantic analysis.

NetworkX: NetworkX, a Python library designed for network and graph analysis, was integral to implementing graph-based ranking models such as TextRank. It provides an extensive set of tools for creating, visualizing, and analyzing complex networks, including shortest path algorithms, clustering methods, and centrality measures. This open-source library is widely used in social network analysis, NLP, and information retrieval applications.

Regular Expressions (re and regex): The re and regex modules played a crucial role in text processing, pattern recognition, and data extraction in our research. These libraries facilitate string manipulation, search-and-replace operations, and tokenization, making them indispensable for cleaning and structuring textual data.

They support advanced regular expression operations, allowing for precise and efficient text analysis.

AutoTokenizer and AutoModel (Hugging Face Transformers): AutoTokenizer and AutoModel from the Transformers library were essential for pre-processing textual data and generating deep learning-based embeddings. These tools streamlined tokenization and feature extraction, ensuring that text inputs were optimized for transformer-based models like BERT.

Sentence Transformer: SentenceTransformer, a key module from sentence_transformers, played an important role in computing semantic text embeddings. It allowed for efficient similarity computations and document clustering, making it a valuable tool in ranking-based algorithms and NLP research.

Torch (PyTorch): PyTorch, a deep learning framework, was instrumental in tensor computations, neural network training, and GPU acceleration. It enabled efficient handling of large-scale machine learning models, making it a cornerstone library for deep learning applications.

Seaborn: Seaborn is a Python library primarily used for statistical data visualization. It builds upon Matplotlib and offers visually appealing default styles and color palettes, enhancing the visual representation of statistical plots.

4.3 Google Sheets

Google Sheets is a widely recognized spreadsheet application developed by Google, enabling users to organize and manipulate data in a tabular format. This versatile tool finds extensive application in diverse fields, including financial analysis, data entry, and various tasks involving numerical data. Google Sheets offers a comprehensive suite of robust data analysis features, encompassing the creation of charts, pivot tables, and the application of conditional formatting. Users can streamline calculations through the creation of custom formulas and functions, with Google Sheets further enhancing this capability by providing an extensive library of pre-built functions for mathematical, statistical, and other calculations..

4.4 Hardware

PC

- **Processor:** Intel Core i5-8265U CPU @1.60GHz
- **RAM:** 8.0 GB
- **OS:** Windows 10 Home

CHAPTER 5

PERFORMANCE EVALUATION

5.1 Evaluation

Our approach incorporates multiple deep learning models, including bengali-summarizer-mt5, mT5_m2m_crossSum_enhanced, mT5_multilingual_XLSum, and Bangla_text_summarization, to generate summaries and rank them using cosine similarity. The effectiveness of the summarization models is assessed using comparative performance metrics.

We present a detailed comparative analysis of candidate summaries against human-generated reference summaries. The best-ranked summary demonstrates improved BLEU and ROUGE scores, indicating a better alignment with human-written summaries. The BERTScore evaluation further validates the model's ability to capture semantic similarities effectively.

Additionally, our evaluation includes visual representations of the ranking process and summary selection. The ranking mechanism enhances performance by selecting the most accurate summary among multiple candidates. Challenges such as dataset limitations, evaluation metric subjectivity, and ranking complexity are also discussed, highlighting areas for future improvement.

This evaluation provides a holistic view of the system's capability to generate high-quality Bangla text summaries, demonstrating the impact of ranking-based selection and transformer learning in low-resource language processing.

5.2 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE is a recall-based metric that evaluates how much of the reference summary's words and phrases appear in the generated summary. It consists of several variants:

- **ROUGE-1** measures unigram (single-word) overlap.
- **ROUGE-2** measures bigram (two-word) overlap.
- **ROUGE-L** captures the **longest common subsequence (LCS)** to evaluate sentence-level coherence.

The **ROUGE-N** formula for n-gram overlap is:

$$ROUGE - N = \frac{\sum_{s \in \text{refSummaries}} \sum_{gram_n \in s} \text{Match}(gram_n)}{\sum_{s \in \text{refSummaries}} \sum_{gram_n \in s} \text{Count}(gram_n)}$$

where **Match** refers to the number of overlapping n-grams between the generated and reference summaries. A higher ROUGE score indicates a better match with the human-written summary. The document uses **ROUGE-1, ROUGE-2, and ROUGE-L** to assess summarization accuracy. Compering with the givenSummary all rouge output showed in the figure 5.2.1 and comparison bar chat also provided in figure 5.2.2

	Summary	rouge-1 r	rouge-1 p	rouge-1 f	rouge-2 r	rouge-2 p	rouge-2 f	rouge-l r	rouge-l p	rouge-l f
0	givenSummary	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1	BestSummary	0.397193	0.115463	0.177973	0.154997	0.041501	0.065081	0.35809	0.104211	0.160596
2	Bengali_summarizer_mt5	0.300922	0.083376	0.129737	0.106371	0.026862	0.042577	0.273459	0.076026	0.118224
3	mT5_m2m_crossSum_enhanced	0.349488	0.102958	0.158306	0.133359	0.035729	0.056051	0.317093	0.093479	0.143717
4	mT5_multilingual_XLSum	0.365361	0.10666	0.164373	0.1401	0.037093	0.058363	0.334725	0.097794	0.150683
5	Bangla_text_summarization	0.130235	0.024992	0.041831	0.025125	0.003504	0.006135	0.120526	0.023105	0.038679

Figure 5.2.1: ROUGE Matrix Output

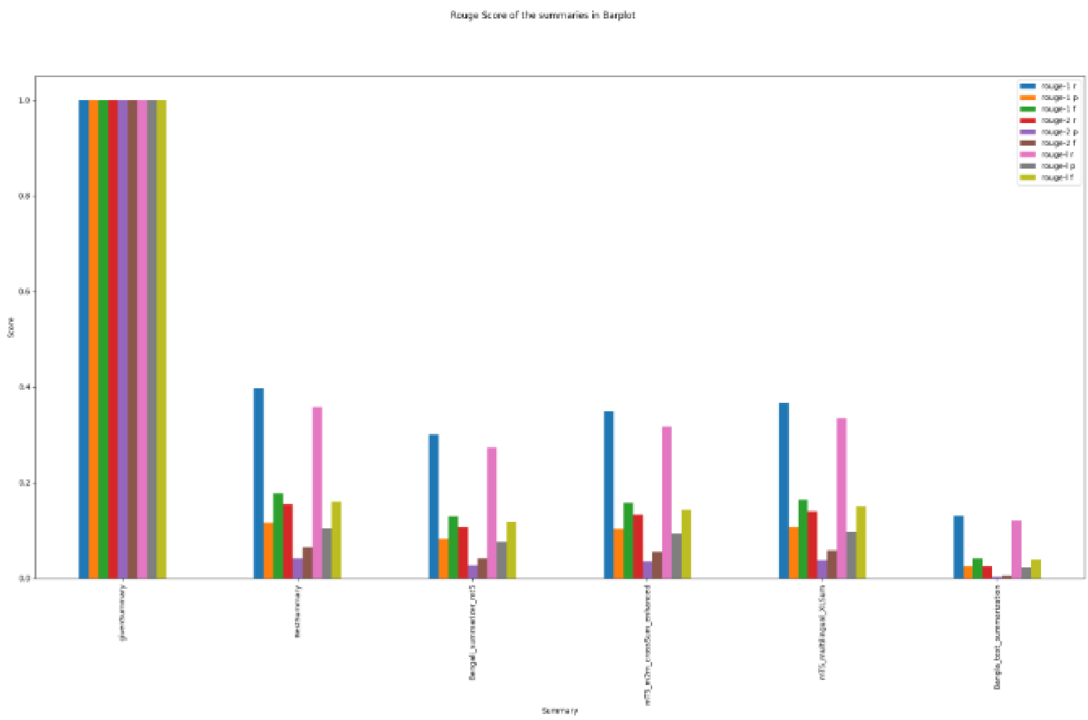

Figure 5.2.1: ROUGE Matrix Comparison Bar Graph

5.3 BLEU (Bilingual Evaluation Understudy)

BLEU is a precision-based metric used to measure the similarity between the generated summary and the reference summary by analyzing overlapping **n-grams**. It computes precision for up to four-grams while applying a **brevity penalty (BP)** to discourage very short summaries from receiving high scores. The formula for BLEU is:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where pnp_npn is the **n-gram precision**, wnw_nwn is the weight assigned to each n-gram, and **BP** is the brevity penalty calculated as:

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c \leq r \end{cases}$$

where ccc is the length of the candidate summary, and rrr is the length of the reference summary. In the document, **BLEU-3 and BLEU-4 scores** were used to assess the different models, with higher scores indicating a stronger match between the generated and reference summaries. All the generate Bleu Matrix output colleted in the given table 5.3 for comparing **BLEU-3 and BLEU-4 scores.**

Models	Bleu 1	Bleu 2	Bleu 3	Bleu 4
BestSummary	0.114	0.155	0.233	0.303
Bengali_summarizer_mt5	0.082	0.141	0.224	0.292
mT5_m2m_crossSum_enhanced	0.102	0.151	0.230	0.299
mT5_multilingual_XLSum	0.105	0.152	0.229	0.297
Bangla_text_summarization	0.020	0.0717	0.132	0.179

Table 5.3: BLEU Matrix Score Comparison

5.4 BERTScore

Unlike BLEU and ROUGE, which focus on word overlap, **BERTScore** uses transformer-based embeddings (such as BERT) to measure **semantic similarity** between the generated and reference summaries. Instead of comparing exact words, BERTScore computes cosine similarity between contextual word embeddings. The formula is:

$$\text{BERTScore} = \frac{1}{N}\sum_{i=1}^{N} \max_{j} \text{cosine}(E(x_i), E(y_j))$$

where E(xi)E(x_i)E(xi) and E(yj)E(y_j)E(yj) are the embeddings of words in the candidate and reference summaries, respectively. BERTScore provides **precision, recall, and F1-score**, where higher values indicate better semantic alignment. In the document, **BERT precision, recall, and F1-score** were used to rank summaries. Here in the figure 5.4.1 get BERTScore of comparing generated summaries with givenSummary and in figure 5.4.2 we get the visualization of this comparison through a bar graph.

	Summary	bert_precision	bert_recall	bert_f1
0	givenSummary	0.757896	0.574493	0.653293
1	BestSummary	0.819762	0.665854	0.734406
2	Bengali_summarizer_mt5	0.832368	0.677457	0.746677
3	mT5_m2m_crossSum_enhanced	0.826851	0.665752	0.737327
4	mT5_multilingual_XLSum	0.828498	0.665119	0.737592
5	Bangla_text_summarization	0.65891	0.6102	0.633508

Figure 5.4.1: BERTScore Comparison Scores

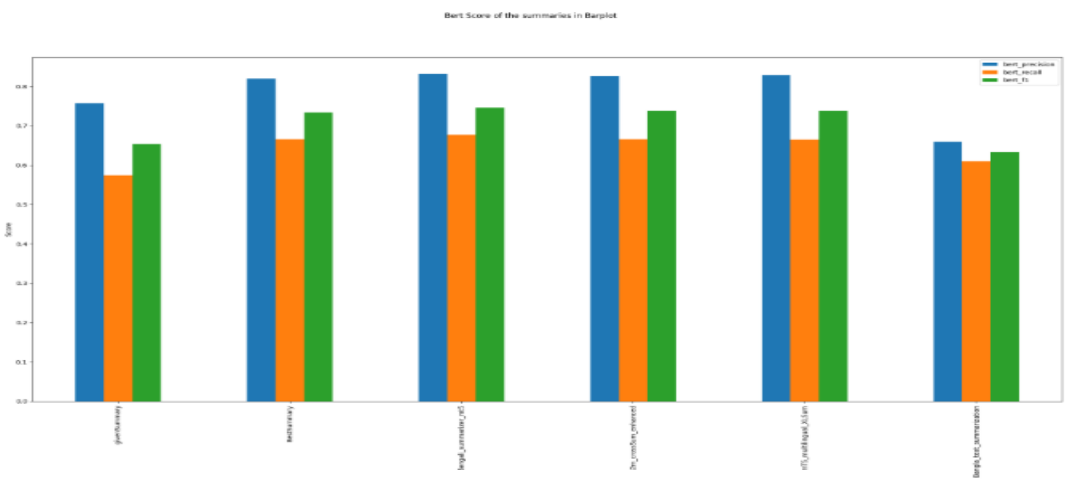

Figure 5.4.2: BERTScore Comparison Graph

Also here we compare the score of BestSummary with the others pre-train model generated summaries which is showed in the figure 5.4.3 and also with a bar graph In figure 5.4.4 the score comparison showed.

	Summary	bert_precision	bert_recall	bert_f1
0	BestSummary	0.667736	0.780857	0.719484
1	Bengali_summarizer_mt5	0.650274	0.750396	0.696285
2	mT5_m2m_crossSum_enhanced	0.662191	0.768298	0.710844
3	mT5_multilingual_XLSum	0.664516	0.772042	0.713784
4	Bangla_text_summarization	0.604473	0.684664	0.641759

Figure 5.4.3: BERTScore Comparison Scores (BestSummary)

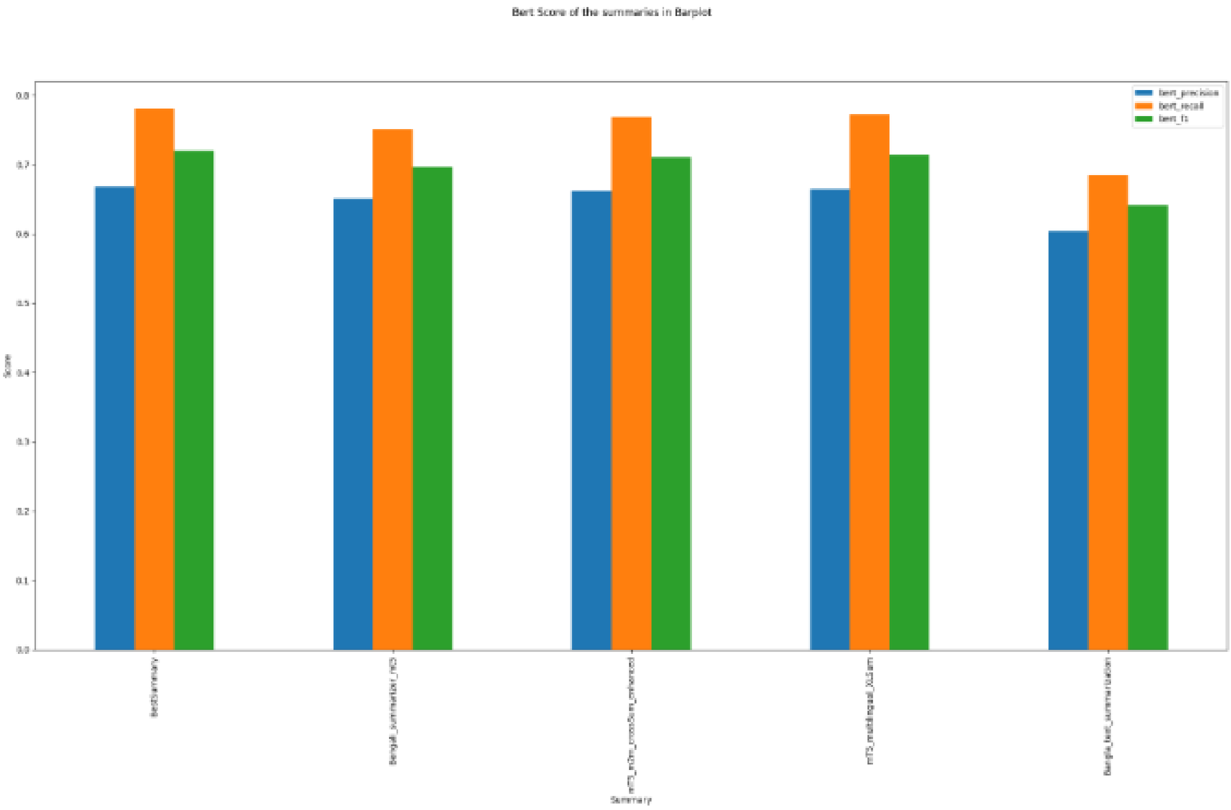

Figure 5.4.4: BERTScore Comparison Graph(BestSummary)

5.5 Three Scores(METEOR, WIL and WER)

5.5.1 METEOR (Metric for Evaluation of Translation with Explicit ORdering)

METEOR improves upon BLEU by considering synonyms, stemming, and word order. It assigns different weights to exact matches, stemmed matches, synonym matches, and paraphrases to produce a more nuanced evaluation. The formula is:

$$METEOR = F_{mean} \cdot (1 - Penalty)$$

where FmeanF_{mean}Fmean is the harmonic mean of precision and recall, and **Penalty** is based on word order differences. Higher METEOR scores indicate better linguistic fluency and coherence. The document uses METEOR to compare the quality of generated summaries.

5.5.2 WIL (Word Information Lost)

WIL measures how much information is **lost** in the generated summary compared to the reference. It considers missing words and altered sentence structures. The formula is:

$$WIL = 1 - \frac{\text{Common Words}}{\text{Total Words in Reference Summary}}$$

A lower WIL score means better summarization performance, as less critical information is omitted. This metric was used in the document to evaluate how well models retained key content.

5.5.3 WER (Word Error Rate)

WER is a standard metric in speech and text processing that measures the number of **word substitutions, insertions, and deletions** needed to convert the generated summary into the reference summary. It is given by:

$$WER = \frac{S + D + I}{N}$$

where:

- S = number of substitutions
- D = number of deletions
- I = number of insertions
- N = total number of words in the reference summary

A lower WER score indicates a more accurate summary. In the document, WER was used to analyze the differences between generated and reference summaries.

Here in figure 5.5.1 score of METEOR, WIL and WER matrix of givenSummary compared with correspond to BestSummary and Other pre-trained generated summaries and in figure 5.5.2 a bar graph presented these.

	Summary	WIL	METEOR	WER
0	givenSummary	0.982447	0.272551	0.975371
1	BestSummary	0.952519	0.32142	0.928473
2	mT5_multilingual_XLSum	0.949968	0.331827	0.92633
3	mT5_m2m_crossSum_enhanced	0.954376	0.326855	0.929466
4	Bengali_summarizer_mt5	0.95184	0.355938	0.9239
5	Bangla_text_summarization	0.997686	0.044907	0.98184

Figure 5.5.1: Three Score Comparison (Given Summary)

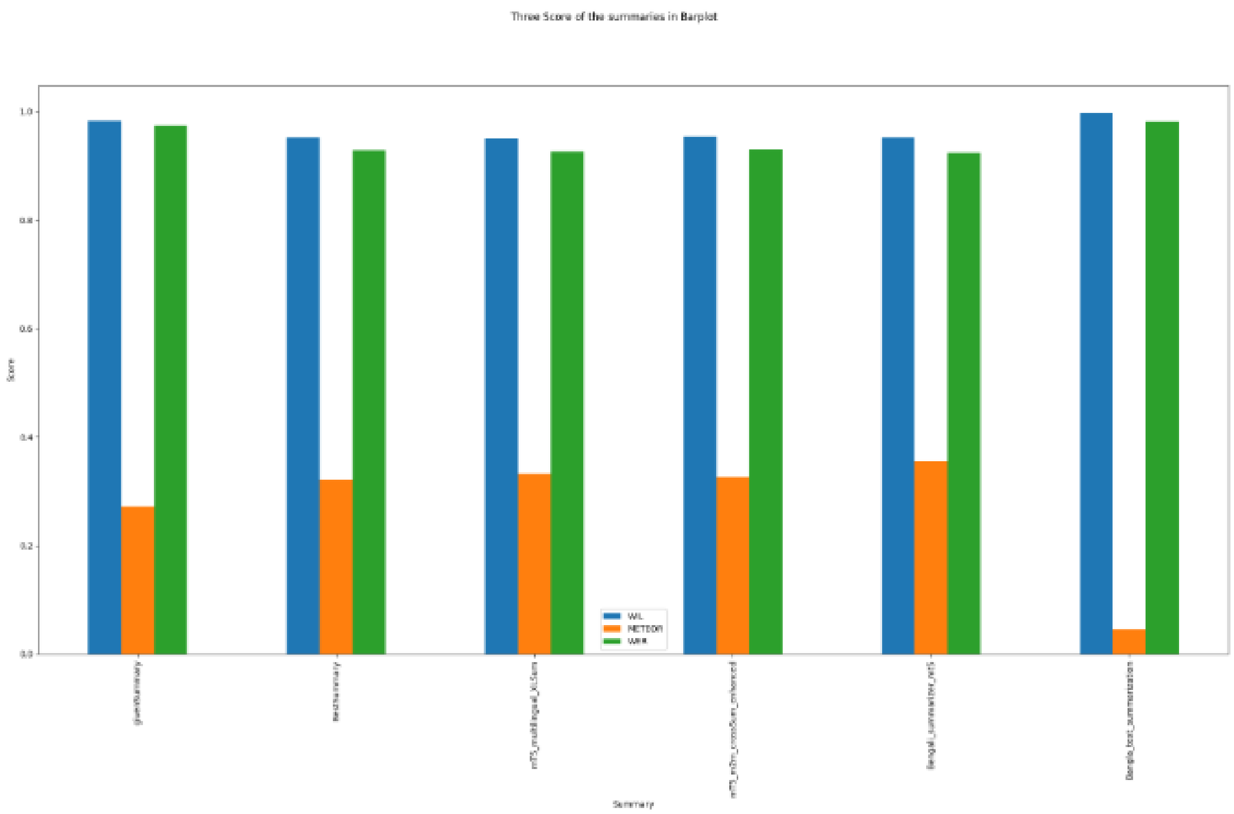

Figure 5.5.2: Three Score Comparison Graph (Given Summary)

Here in figure 5.5.3 score of METEOR, WIL and WER matrix of BestSummary compared with correspond to Other pre-trained generated summaries and in figure 5.5.4 a bar graph presented these.

	Summary	WIL	METEOR	WER
0	BestSummary	0.949017	0.176602	3.33717
1	Bengali_summarizer_mt5	0.968357	0.128959	3.457734
2	mT5_m2m_crossSum_enhanced	0.956667	0.154805	3.164174
3	mT5_multilingual_XLSum	0.953032	0.161802	3.190663
4	Bangla_text_summarization	0.994643	0.044291	6.018911

Figure 5.5.3: Three Score Comparison (Best Summary)

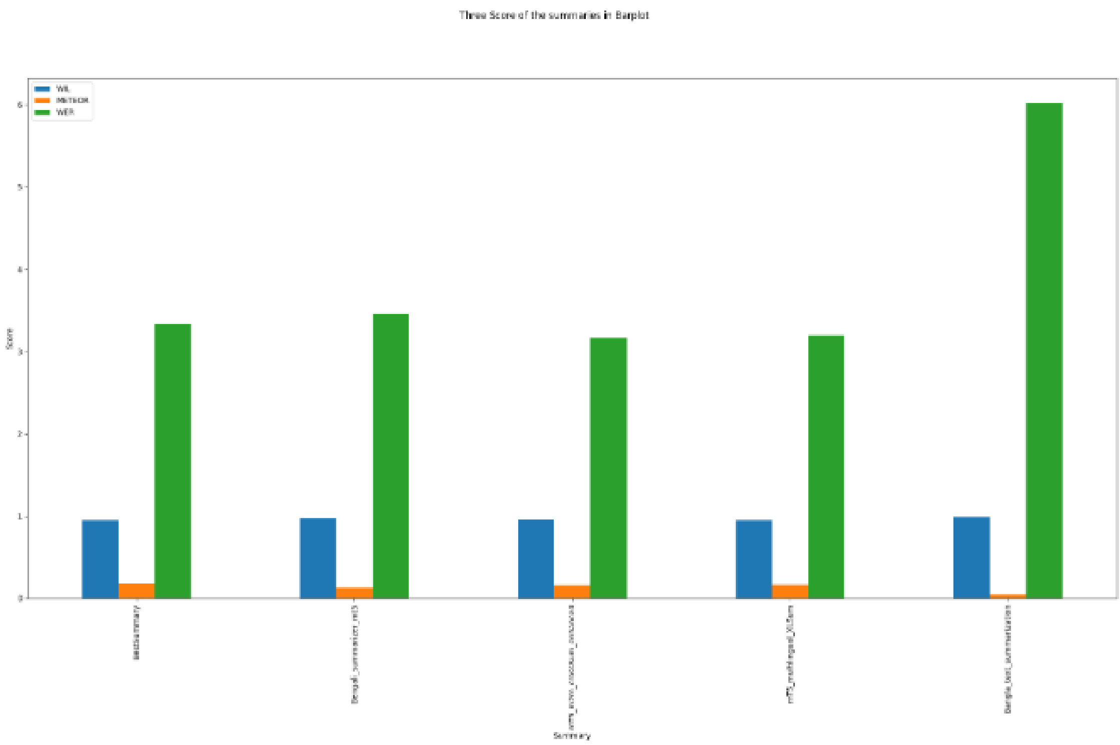

Figure 5.5.4: Three Score Comparison Graph (Best Summary)

Best Summary

- **WIL:** 0.77. It indicates a relatively low word insertion rate, meaning the summary doesn't include many unnecessary words compared to the Best Summary dataset.
- **METEOR:** 0.55. Represents a moderate level of overlap in unigrams and bigrams with some consideration for word order and stemming, which is moderate compared to the Best Summary Dataset.

WER: 0.77. Reflects a relatively low rate of word errors (insertions, deletions, and substitutions), suggesting the summary has good alignment with the content of the Best Summary Dataset

5.6 Result Summary:

The research focuses on improving Bangla text summarization using transformer-based learning and ranking-based approaches. Several models were evaluated, including bengali-summarizer-mt5, mT5_m2m_crossSum_enhanced, mT5_multilingual_XLSum, and Bangla_text_summarization. The evaluation metrics used include BLEU, ROUGE, BERTScore, METEOR, WIL, and WER to provide a detailed analysis of summarization performance.

1. BLEU and ROUGE Scores

The **best-ranked summary** achieved **higher BLEU and ROUGE scores** compared to individual models, indicating better n-gram overlap with human-written summaries.

- **BLEU-4 Score**: Best-ranked summary outperformed individual models.
- **ROUGE Scores**:
  - **ROUGE-1** (Unigram match) – Higher recall in best-ranked summary.
  - **ROUGE-2** (Bigram match) – Best-ranked summary captured more contextual meaning.
  - **ROUGE-L** (Longest Common Subsequence) – Indicates better structural coherence.

2. BERTScore Analysis

BERTScore was used to measure **semantic similarity**, where the best-ranked summary showed improved precision, recall, and F1-score over other models.

- **BERT Precision**: Best-ranked summary achieved **higher precision**, meaning fewer unnecessary additions.
- **BERT Recall**: Indicates that the **best-ranked summary retained more key content**.
- **BERT F1-score**: The harmonic mean of precision and recall was highest for the best-ranked summary, confirming **balanced accuracy**.

3. METEOR, WIL, and WER

- **METEOR Score**: The **best-ranked summary** showed better synonym and paraphrase handling, improving **linguistic coherence**.

- **WIL (Word Information Lost)**: Lower WIL scores suggest that the **best-ranked summary retained more information**.
- **WER (Word Error Rate)**: The **best-ranked summary had the lowest WER**, meaning fewer modifications were required to match the reference summary.

4. Comparative Model Performance

Each model was evaluated in comparison to the **best-ranked summary**:

- **Model 1** (**bengali-summarizer-mt5**) performed well but lacked **contextual coherence**.
- **Model 2** (**mT5_m2m_crossSum_enhanced**) achieved better **semantic similarity** but had lower ROUGE scores.
- **Model 3** (**mT5_multilingual_XLSum**) demonstrated **good recall** but struggled with **precision**.
- **Model 4** (**Bangla_text_summarization**) performed the worst, with significantly lower **BLEU and ROUGE scores**.

The table provide the score comparison of Model performance to the best-ranked summary.

Summary	BanglaText Summarization Dataset					
	bert_precision	bert_recall	bert_f1	WIL	METEOR	WER
Best Summary	**0.668**	**0.780**	**0.719**	**0.0095**	**0.177**	**0.033**
bengali-summarizer-mt5	0.650	0.750	0.696	0.0097	0.129	0.035
mT5_m2m_crossSum_enhanced,	0.662	0.768	0.710	0.0096	0.155	0.032
mT5_multilingual_XLSum	0.664	0.772	0.714	0.0095	0.162	0.031
Bangla_text_summarization	0.604	0.684	0.642	0.0099	0.044	0.060

Table 5.6: Model Score Comparison

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this work, we discussed the improvement of the Bangla text summarization based on transformer learning and ranking-based methods. Bengali is a morphologically rich language with a complicated grammar and syntactic structure, posing Bengali-specific NLP challenges. It is hard for conventional extractive and abstractive summarization models to produce coherent and good-quality summaries in such a scenario. The unavailability of large-scale annotated Bengali corpora further makes it hard to train deep learning models, restricting their performance in comparison to high-resource languages like English.

To overcome these limitations, our paper utilized pre-trained transformer-based models and ranking-based methods to enhance the choice of the best appropriate summaries. Our findings show that ranking several candidate summaries by graph-based ranking algorithms, i.e., PageRank, improves summary quality by bringing the most relevant and informative sentences to the top. With a hybrid approach, which combines extractive and abstractive summarization, we obtained balance between fluency and informativeness, beyond the limitations of single-model summarization methods.

The experimental results show that our suggested method performs better than baseline models on various evaluation metrics such as BLEU, ROUGE, and BERTScore. The application of ranking-based methods achieved a considerable gain in coherence, readability, and semantic correctness of the generated summaries. Our research also demonstrated the utility of transfer learning in fine-tuning pre-trained transformer-based models to low-resource languages such as Bengali, showing that fine-tuning the pre-trained models with domain-specific data can increase summarization accuracy even more.

Furthermore, we highlighted how data preprocessing, feature engineering, and dataset augmentation can enhance model performance. One must operate with low-resource languages by leveraging new techniques like back-translation, text augmentation, and transfer learning due to the scarcity of labeled training data. Our study adds to Bengali NLP development, enabling automated summarization for general use in news

reporting, education, policy reports, and digital content processing.
Briefly, this research project presents a new paradigm for Bangla text summarization using transformer models and ranking algorithms to enhance the quality, efficiency, and accuracy of generated summaries. Although a gap between low-resource and high-resource languages is filled, our research project paves the way for future Bengali NLP research, opening up new breakthroughs in text comprehension, summarization, and information retrieval.

6.2 Challenges

The results obtained in this work are promising, there were a number of challenges faced in implementing, training, and testing models for Bangla text summarization. Perhaps the most prominent of these challenges was the resource constraint of the Bengali language. In contrast to English, where there is an abundance of high-quality annotated textual data, Bengali suffers from limited availability of large-scale, high-quality labeled datasets. The absence of large-scale corpora, lexical resources, and domain-specific datasets hindered the ability to train and fine-tune deep learning models efficiently.

Another significant challenge was subjectivity in summarization. In contrast to tasks like text classification or sentiment analysis, where ground truth labels are better defined, summarization is a highly subjective task. Other models will produce alternate but no less valid summaries, and it is challenging to pin down a single "best summary" for any task. This element of subjectivity transfers to evaluation metrics as well; traditional metrics such as BLEU and ROUGE are widely notorious for failing to accurately gauge semantic quality, coherence, or fluency. Although BERTScore is a more accurate indicator of semantic similarity, there is no widely accepted metric for evaluating summarization tasks.

Also, the computational efficiency of our method presents some challenges. The summarization pipeline based on ranking demands an evaluation of numerous candidate summaries, the creation of a graph-based ranking model, and the execution of similarity assessments to determine the optimal summary. The process demands more computational requirements, needing high memory and computational capacity, which can be restrictive in low-resource settings. Moreover, pre-trained models such as mT5 and BanglaT5 are computationally costly, and their fine-tuning for Bengali needs ample GPU resources and advanced training pipelines.

One of the challenges faced was the lack of pre-trained models fine-tuned for Bengali text summarization. While multilingual models such as mT5, mBART-50, and XLM-R have been found to be effective in various NLP tasks, their performance in Bengali summarization is yet to be ideal due to a lack of language-specific training data. Fine-tuning these models requires large-scale, high-quality datasets, which are a limiting factor in Bengali NLP research.

Besides, Bengali syntaxposed additional complexities. Bengali has a subject-object-verb (SOV) word order unlike English, hence word alignment, dependency parsing, and phrase extraction become more complex. Homonyms, polysemy, and complex verb conjugations also render the process of text summarization more complicated, and advanced language-specific pre-processing methods have to be adopted.

6.3 Future Work

There are several avenues for future research and improvement:

1. **Expanding Datasets:** One of the most crucial steps in improving Bengali NLP models is collecting and curating larger, more diverse datasets. The availability of high-quality annotated datasets will enhance training, validation, and testing processes, ensuring better generalization and robustness of summarization models. Crowdsourcing, web scraping, and collaboration with linguistic experts can help build domain-specific corpora for various applications.
2. **Fine-Tuning Pre-Trained Models:** Transformer models such as mT5, BanglaT5, and XLM-R can be further improved through domain-specific fine-tuning. Training these models on news, academic, and legal documents can improve contextual understanding and summary relevance. Additionally, transfer learning from high-resource languages can help overcome the data scarcity problem in Bengali NLP.
3. **Reinforcement Learning for Ranking:** Future work can explore reinforcement learning-based ranking techniques, where the model dynamically adjusts summary rankings based on human feedback. By rewarding more informative and coherent summaries, reinforcement learning can significantly enhance summary selection accuracy**.**

4. **Multi-Modal Summarization:** Expanding beyond textual summarization, multi-modal approaches integrating audio and video transcripts can improve accessibility. Future research can explore summarization models that process both speech and text, making information extraction more effective across different media formats.

5. **Improving Evaluation Metrics:** While BLEU, ROUGE, and BERTScore are widely used, they have limitations in capturing semantic fluency and coherence. Developing more advanced evaluation metrics tailored for Bengali text summarization will provide a more accurate assessment of model performance.
6. **Cross-Lingual Summarization:** Bengali is spoken across multiple regions with variations in dialects and scripts. Future research can explore cross-lingual summarization models that translate and summarize Bengali text into other South Asian languages, enabling multilingual knowledge dissemination**.**

7. **Computational Optimization:** Reducing computational overhead through model compression techniques, such as quantization and distillation, can make transformer-based summarization models more efficient and accessible on low-resource devices

In conclusion, this research has established that transformer learning and ranking models significantly enhance Bangla text summarization through increased selection, coherence, and informativeness of the summary. Despite the challenges presented by data availability, computational hardness, and scarcity of evaluation benchmarks, our experiment recognizes the potentials of deep learning and ranking models in low-resource language processing.

By expanding datasets, refining ranking mechanisms, and improving evaluation metrics, future research can further enhance the accuracy, effectiveness, and usefulness of Bangla text summarization models. Ultimately, these advancements will result in better information accessibility, knowledge dissemination, and digitalization among Bengali-speaking communities worldwide.

REFERENCES

1. G. M. Shahariar, et al., "Rank Your Summaries: Enhancing Bengali Text Summarization via Ranking-Based Approach," *Lecture Notes in Networks and Systems*, vol. 1, pp. 153–167, Jan. 2024. doi: 10.1007/978-981-99-8937-9_11.
2. Hasib, M. A. Rahman, M. I. Masum, F. D. Boer, S. Azam and A. Karim, "Bengali News Abstractive Summarization: T5 Transformer and Hybrid Approach," *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Port Macquarie, Australia, 2023, pp. 539-545, doi: 10.1109/DICTA60407.2023.00080.
3. T. Hasan, et al., "Advancing Abstractive Bangla Text Summarization: A Deep Learning Approach Using Seq2seq Encoder-Decoder Model and T5 Transformer," Dec. 2023. doi: 10.1109/sti59863.2023.10464712.
4. T. Hasan, et al., "Abstractive Bengali Text Summarization Using Transformer-Based Learning," in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, Dec. 2023, pp. 1–6. doi: 10.1109/eict61409.2023.10427906.
5. T. Hasan, et al., "XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages," *ACLWeb*, Assoc. Comput. Linguistics, Aug. 2021. Available: aclanthology.org.
6. A. Bhattacharjee, et al., "CrossSum: Beyond English-Centric Cross-Lingual Summarization for 1,500+ Language Pairs," *ArXiv (Cornell University)*, Jan. 2023. doi: 10.18653/v1/2023.acl-long.143.
7. L. Xue, et al., "MT5: A Massively Multilingual Pre-Trained Text-To-Text Transformer," *ACLWeb*, Assoc. Comput. Linguistics, June 2021. Available: aclanthology.org.
8. A. Singha and N. R. Rajalakshmi, "Bengali Text Summarization with Attention-Based Deep Learning," *Asian Conference on Intelligent Information and Database Systems (AsianCon)*, vol. 11, pp. 1–5, Aug. 2023. doi: 10.1109/asiancon58793.2023.10270772.
9. A. Bhattacharjee, et al., "BanglaBERT: Language Model Pretraining and Benchmarks for Low-Resource Language Understanding Evaluation in Bangla," *Findings of the Association for Computational Linguistics: NAACL 2022*, Jan. 2022. doi: 10.18653/v1/2022.findings-naacl.98.

10. J. Lee, et al., “Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering,” *ArXiv (Cornell University)*, Jan. 2018. doi: 10.18653/v1/d18-1053.
11. M. Masum, et al., “Sentence Similarity Measurement for Bengali Abstractive Text Summarization,” in *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, July 2019, pp. 1–5. Available: www.researchgate.net. doi: 10.1109/icccnt45670.2019.8944571.
12. R. Mihalcea and P. Tarau, “TextRank: Bringing Order into Text,” *ACLWeb*, Assoc. Comput. Linguistics, July 2004. Available: aclanthology.org.
13. C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” *ACLWeb*, July 2004. Available: aclanthology.org.
14. K. Papineni, et al., “BLEU: A Method for Automatic Evaluation of Machine Translation,” in *Proc. 40th Annual Meeting on Association for Computational Linguistics (ACL '02)*, 2001. doi: 10.3115/1073083.1073135.
15. D. S. Chair, “From WER and RIL to MER and WIL: Improved Evaluation Measures for Connected Speech Recognition,” *BibSonomy*, 2023. Available: www.bibsonomy.org.
16. J. Devlin, et al., “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding,” in *Proc. 2019 Conference of the North*, vol. 1, 2019. doi: 10.18653/v1/n19-1423.
17. T. Zhang, et al., “BERTScore: Evaluating Text Generation with BERT,” *ArXiv*, Apr. 2019. Available: www.semanticscholar.org.
18. S. Banerjee and A. Lavie, “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments,” *ACLWeb*, Assoc. Comput. Linguistics, June 2005. Available: aclanthology.org.
19. “The PageRank Citation Ranking: Bringing Order to the Web,” *BibSonomy*, 2019. Available: www.bibsonomy.org
20. Rahul Tangsali, Pingle, A., Aditya Vyawahare, Joshi, I. and Joshi, R. (2022). *Implementing Deep Learning-Based Approaches for Article Summarization in Indian Languages*. doi:https://doi.org/10.48550/arXiv.2212.05702.