

**Department of Electrical and Computer Engineering**

**North South University**

**Directed Research**

**“**Bengali News Summarization Using Multilingual Pre-Trained Models for Improved Text Generation**”**

**MOHAMMED NAHID HOSSAIN ID# 2011964042**

**MD. ASHRAFUL BHUYAN SHUVON ID# 1931405642**

**Md. ASIF IQBAL ID# 2012940642**

**JANNATUL ISLAM ID# 2014159642**

**Faculty Advisor:**

**Intisar Tahmid Naheen**

**Senior Lecturer**

**ECE Department**

**Spring, 2023**

# APPROVAL

MOHAMMED NAHID HOSSAIN (ID # 2011964042), MD. ASHRAFUL BHUYAN SHUVON (ID # 1931405642), Md. ASIF IQBAL (ID # 2012940642), and JANNATUL ISLAM (ID # 2014159642) from Electrical and Computer Engineering Department of North South University, have worked on the Directed Research Project titled “**Bengali News Summarization Using Multilingual Pre-Trained Models for Improved Text Generation**” under the supervision of Dr. Riasat Khan partial fulfillment of the requirement for the degree of Bachelors of Science in Engineering and has been accepted as satisfactory.

**Supervisor’s Signature**

…………………………………….

**Intisar Tahmid Naheen**

**Senior Lecturer**

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

**Chairman’s Signature**

…………………………………….

**Dr. Mohammad Abdul Matin**

**Professor**

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

# DECLARATION

This is to declare that this project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. All project related information will remain confidential and shall not be disclosed without the formal consent of the project supervisor. Relevant previous works presented in this report have been properly acknowledged and cited. The plagiarism policy, as stated by the supervisor, has been maintained.

Students’ names & Signatures

**1. MOHAMMED NAHID HOSSAIN**

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

**2. MD. ASHRAFUL BHUYAN SHUVON**

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

**3. Md. ASIF IQBAL**

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

**4. JANNATUL ISLAM**

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

# ACKNOWLEDGEMENTS

The authors would like to express their heartfelt gratitude towards their project and research supervisor, Intisar Tahmid Naheen, Senior Lecturer, Department of Electrical and Computer Engineering, North South University, Bangladesh, for his invaluable support, precise guidance and advice pertaining to the experiments, research and theoretical studies carried out during the course of the current project and also in the preparation of the current report.

Furthermore, the authors would like to thank the Department of Electrical and Computer Engineering, North South University, Bangladesh for facilitating the research. We would also like to thank my friends Mohammed Nahid Hossain, Ashraful Bhuyan Shuvon, Md. Asif Iqbal, and Jannatul Islam for helping us in this project. The authors would also like to thank their loved ones for their countless sacrifices and continual support.

# ABSTRACT

“Bengali News Summarization Using Multilingual Pre-Trained Models for Improved Text Generation”

Start your Abstract here. All papers must include an abstract. Abstract should contain the main work of the project in concise paragraph(s). Do not cite references or use abbreviations in the abstract. Abstract will be copied and included in the conference proceedings or journal’s webpage or department website. You should use Times New Roman Font with 12 Font size. You should use Justify Paragraph option. You should use 1.5 Line Spacing. Do not use References, Symbols, Special Characters, Footnotes, or Math equations in the Abstract. Write few sentences from each of the following: introducing the topic, current status, methods followed, results acquired and impact/significance of the result.

# TABLE OF CONTENTS

[APPROVAL ii](#_Toc190104823)

[DECLARATION iii](#_Toc190104824)

[ACKNOWLEDGEMENTS iv](#_Toc190104825)

[ABSTRACT v](#_Toc190104826)

[TABLE OF CONTENTS vi](#_Toc190104827)

[LIST OF FIGURES viii](#_Toc190104828)

[LIST OF TABLES ix](#_Toc190104829)

[Chapter 1 Introduction 1](#_Toc190104830)

[1.1 Background and Motivation 1](#_Toc190104831)

[1.2 Purpose and Goal of the Project 2](#_Toc190104832)

[1.3 Organization of the Report 3](#_Toc190104833)

[Chapter 2 Research Literature Review 4](#_Toc190104834)

[2.1 Existing Research and Limitations 4](#_Toc190104835)

[Chapter 3 Methodology 6](#_Toc190104836)

[3.1 System Design 6](#_Toc190104837)

[3.2 Dataset Collection 7](#_Toc190104838)

[3.3 Exploratory Data Analysis 7](#_Toc190104839)

[3.4 Dataset Preprocessing 11](#_Toc190104840)

[3.5 Train Test Split 12](#_Toc190104841)

[3.6 Models 13](#_Toc190104842)

[3.7 Evaluation Metrics 22](#_Toc190104843)

[3.8 Software Implementation 23](#_Toc190104844)

[Chapter 4 Investigation/Experiment, Result, Analysis and Discussion 25](#_Toc190104845)

[4.1 Investigation/Experiment 25](#_Toc190104846)

[4.2 Results 26](#_Toc190104847)

[4.3 Graphs 28](#_Toc190104848)

[4.4 Analysis 30](#_Toc190104849)

[4.5 Discussion 30](#_Toc190104850)

[Chapter 5 Conclusions 31](#_Toc190104851)

[8.1 Summary 31](#_Toc190104852)

[8.2 Limitations 31](#_Toc190104853)

[8.3 Future Improvement 32](#_Toc190104854)

[References 33](#_Toc190104855)

# LIST OF FIGURES

Figure 3.1. System Design Diagram 6

Figure 3.3.1. Train Dataset Length Visualization 7

Figure 3.3.2. Validation Dataset Length Visualization 8

Figure 3.3.3. Test Dataset Length Visualization 8

Figure 3.3.4. Dataset Shape Visualization 9

Figure 3.3.6. Box plot of Article Length 10

Figure 3.3.5. Distribution of Text Lengths (Headline, Summary, Article) 10

Figure 3.3.7. Box plot of Summary Lengths 11

Figure 3.4.1. Missing Values 11

Figure 3.4.2. - Duplicate Values 11

Figure 3.4.3. – Outliers in Article 12

Figure 3.4.4. – Outliers in Summary 12

Figure 3.6.1. – MT5 Model Architecture 15

Figure 3.6.3. – mBart-50 Model Architecture 19

Figure 4.3.1. – Training & Validation Loss of mBart-50 28

Figure 4.3.2. – Training & Validation Loss of BanglaT5 28

Figure 4.3.3. – Training & Validation Loss of mT5-base 29

Figure 4.3.4. – Training & Validation Loss of mT5-small 29

Figure 4.3.5. – Training & Validation Loss of Bengali Llama 30

# LIST OF TABLES

[TABLE I. Parameters used in different models after hyperparameter tuning 25](#_heading=h.z337ya)

[TABLE II. Results of different models on Existing XL-Sum Bengali Dataset 26](#_heading=h.z337ya)

[TABLE III. Results of different models on Custom Scrapped Ittefaq News Dataset 27](#_heading=h.z337ya)

# Chapter 1 Introduction

## Background and Motivation

The rapid expansion of internet content, particularly news, has made it difficult for readers to keep up with current affairs. Thousands of articles are published daily on multiple platforms, making it difficult and time-consuming to skim through each manually. Automatic text summarization provides a solution by offering concise versions of these articles, helping readers grasp the main ideas quickly.

Text summarization is an active field in natural language processing (NLP), and Significant progress has been made in summarizing articles in languages such as English. These advancements have been fueled by the availability of large datasets, powerful machine-learning models, and robust computing resources. However, despite these advancements, text summarization in less-represented languages, such as Bengali, still needs to be explored.

Bengali is spoken by over 230 million people worldwide, making it the seventh most spoken language globally. Despite its wide usage, more NLP tools must be tailored to Bengali, particularly in automatic summarization. The lack of datasets, pre-trained models, and resources specific to Bengali limits the development of effective summarization systems for this language. As Bengali news content grows online, tools are urgently needed to help readers quickly digest important information.

The motivation for this research arises from this gap. With the increasing volume of Bengali news articles, there is a clear need for an automatic summarization tool that can reduce lengthy news articles to manageable summaries. Such a tool would save readers time and provide them with the essential points of an article at a glance. Developing this model could also contribute to the broader field of NLP for low-resource languages, demonstrating the potential for creating sophisticated language models in underrepresented languages like Bengali.

Our project focuses on developing a text summarization model for Bengali news articles to address this gap. We aim to create an efficient summarization tool by leveraging pre-trained models suited explicitly for the Bengali language, such as mBART-50, FLAN-T5, mT5, IndicBART, and BanglaT5. These models have been fine-tuned on both a custom dataset of 40,000 Bengali news articles as well as an existing dataset of Bengali news articles to enhance their performance in summarization tasks. Due to their architecture and multilingual capabilities, these models represent some of the best options for handling text generation and summarization in the Bengali language.

By constructing a large dataset of Bengali news articles and implementing state-of-the-art NLP techniques, this research aims to build a model capable of generating accurate and readable summaries. We aim to provide a valuable resource for readers and the NLP community by advancing automatic text summarization for the Bengali language.

## Purpose and Goal of the Project

The primary purpose of this project is to create a compelling and reliable automatic summarization model for Bengali news articles. We have employed various state-of-the-art pre-trained models highly suited for Bengali text processing, including mBART-50, FLAN-T5, mT5, IndicBART, and BanglaT5. These models were trained and fine-tuned on both a custom dataset and an existing dataset of Bengali news articles to improve their ability to generate accurate and coherent summaries.

The primary goal of this project is to make online news content more accessible for Bengali-speaking readers by developing an effective summarization tool. To achieve this, a large dataset of Bengali news articles, including their corresponding summaries, has been curated, which serves as the foundation for training and fine-tuning pre-trained models. The project focuses on fine-tuning models like mBART-50, FLAN-T5, mT5, IndicBART, and BanglaT5 to handle the unique aspects of Bengali syntax, grammar, and semantics, enabling them to generate meaningful and coherent summaries. The performance of these models will be evaluated using standard metrics like ROUGE scores, ensuring both quantitative and qualitative assessments of summary quality, readability, and coherence. Additionally, this research aims to advance natural language processing (NLP) for low-resource languages by addressing the challenges of Bengali text summarization and offering practical solutions. Ultimately, the project seeks to enhance the efficiency of news consumption for Bengali readers while contributing to the broader field of NLP research for the Bengali language.

## Organization of the Report

This report is organized into several chapters, each focusing on a different aspect of the research. The first chapter provides an introduction, covering the project's background, motivation, purpose, and goals. Chapter two presents a literature review, discussing existing work on text summarization and exploring the use of pre-trained models in natural language processing, focusing on Bengali text. The third chapter outlines the methodology, detailing the dataset collection, preprocessing, and fine-tuning of pre-trained models such as mBART-50, FLAN-T5, mT5, IndicBART, and BanglaT5. Chapter four covers the experimental results and analysis, where the performance of the models is evaluated using metrics like ROUGE, and the challenges encountered during the process are discussed. Finally, chapter five concludes the report, summarizing the essential findings and contributions of the project and providing suggestions for future research and improvements in Bengali text summarization.

# Chapter 2 Research Literature Review

## 2.1 Existing Research and Limitations

Text summarization is critical in natural language processing (NLP) to produce a concise version of a longer text while preserving vital information. Summarization techniques can be classified into two types: extractive and abstractive summarization. Extractive summarization [1] selects significant sentences or phrases from the source text, whereas abstractive summarization generates new sentences that may contain unseen words or phrases, much like a human-written summary. Earlier methods in text summarization were primarily extractive, relying on sentence ranking techniques such as the Term Frequency-Inverse Document Frequency (TF-IDF) and graph-based algorithms like TextRankethods. However, although effective for extracting key sentences, it often fails to help generate coherent and contextually fluent summaries. Abstractive summarization [2], on the other hand, requires deep linguistic understanding and often leverages machine learning techniques, particularly neural networks. In paper [3], The research used the CNN/Daily Mail and New York Times datasets for abstractive summarization, generating oracle summaries through a greedy algorithm that maximizes ROUGE scores. BERTSUM was employed with additional summarization-specific layers to capture document-level features. Instead of a simple classifier, the Inter-sentence Transformer applied more Transformer layers on sentence representations. Trigram blocking was used to reduce redundancy. Ablation studies showed interval segments and trigram blocking improved performance, with BERTSUM + Transformer achieving the best results (ROUGE-1: 43.25, ROUGE-2: 20.24, ROUGE-L: 39.63). This paper [4], used the ILSUM 2022 dataset, containing English, Hindi, and Gujarati news articles, was used to train models on extractive summaries—SentencePiece tokenizer processed texts with padding for uniform sequence lengths. For Gujarati, sentences were translated to English using Google Translate before mapping. Fine-tuned models included PEGASUS, BRIO, T5, and SentenceBERT for English; IndicBART, XL-Sum, and mBART for Hindi; and mBART and XL-Sum for Gujarati. The best ROUGE scores were PEGASUS for English (ROUGE-1: 0.5618), IndicBART for Hindi (ROUGE-1: 0.5536), and Translation + PEGASUS for Gujarati (ROUGE-1: 0.2028). In paper [5], The dataset consists of 2,755 COVID-19 news articles from the Canadian Broadcasting Corporation (CBC), published between January 8 and March 3, 2020, with full texts and summaries. Preprocessing included contraction handling, lowercasing, tokenization, and GloVe word embeddings. The model architecture was a transformer with encoder-decoder layers, dropout, normalization, and GELU activation for better output than ReLU or ELU. Models used were MTDTG 2, MTDTG 5, and MTDTG 6, with MTDTG 6 achieving ROUGE-1 of 0.58 and ROUGE-2 of 0.42. Additionally, MTDTG 6 results were analyzed using a word cloud to represent word frequency from the generated summaries.

# Chapter 3 Methodology

## 3.1 System Design

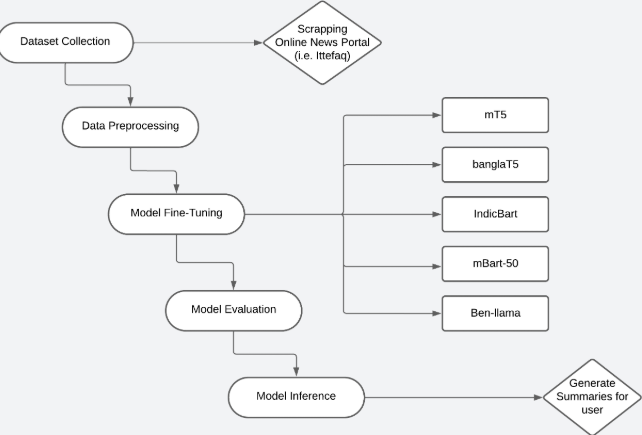


Figure 3.1. System Design

## 3.2 Dataset Collection

We have collected our Bengali news dataset using web scrapping data from one of the oldest newspapers of Bangladesh, The Daily Ittefaq [1]. Each row consists of a title heading, a short summary of the entire news, and finally the entire description of the news. We have extracted the different parts of each news with our custom-made web scrapper. The news portal consists of 12 different sub categories in Bangla news section. Initially we extracted the links of each and every news starting from 1st July 2023 to 30th June 2024, which is mainly a collection of news of the previous one year. Then using the links, we extracted the title, summary, and news description. Finally, our collected dataset consisted of 39,142 of unique data which were ready for further use.

The web scraper is designed to collect news articles from the Ittefaq website, focusing on a specific date range. It automates the browsing process using Selenium and extracts headlines, summaries, and full article text for each news item found. This web scraper effectively automates the process of collecting news articles from the Ittefaq website and performs the data extraction task extremely well. It manages the complexity of web interactions and handles exceptions gracefully, ensuring robust performance even when faced with unexpected webpage structures or loading issues.

## 3.3 Exploratory Data Analysis

3.3.1 – Train Dataset Length Visualization

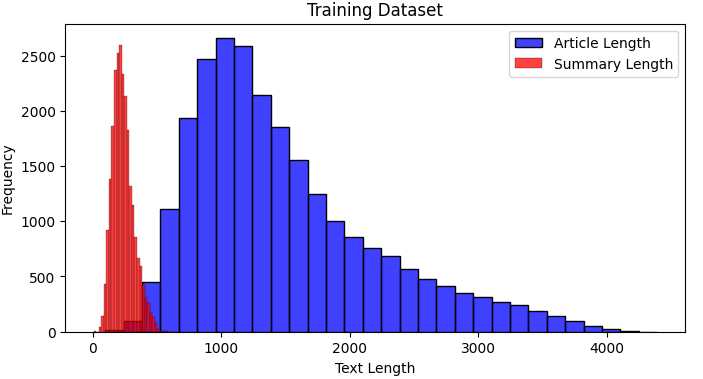


Figure 3.3.1. Train Dataset Length Visualization

3.3.2 – Validation Dataset Length Visualization

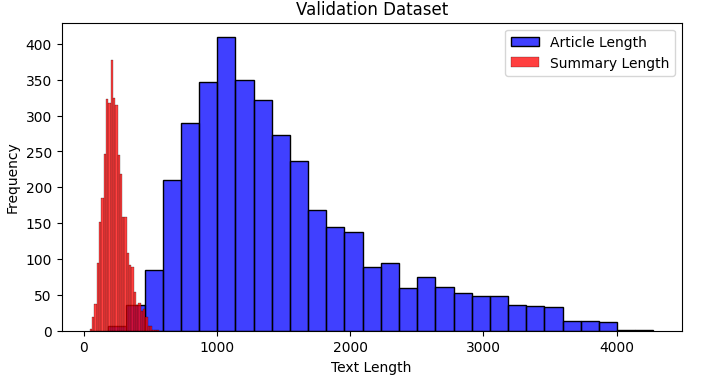
****

Figure 3.3.2. Validation Dataset Length Visualization

3.3.3 – Test Dataset Length Visualization

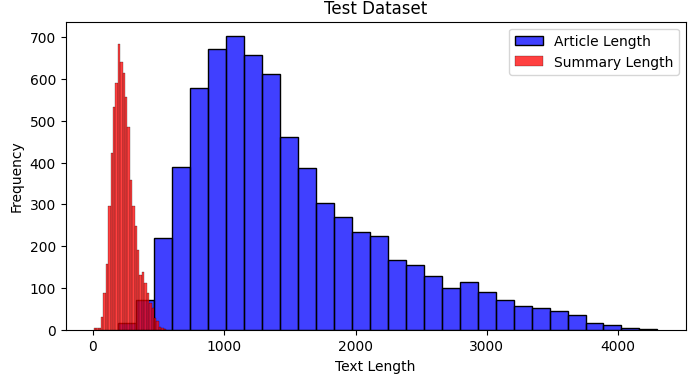


Figure 3.3.3. Test Dataset Length Visualization

3.3.4 – Dataset Shape Visualization

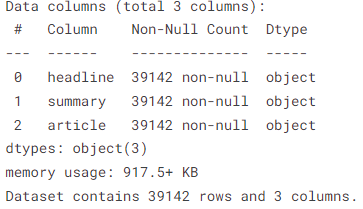


Figure 3.3.4. Dataset Shape Visualization

3.3.5 – Distribution of Text Lengths (Headline, Summary, Article)

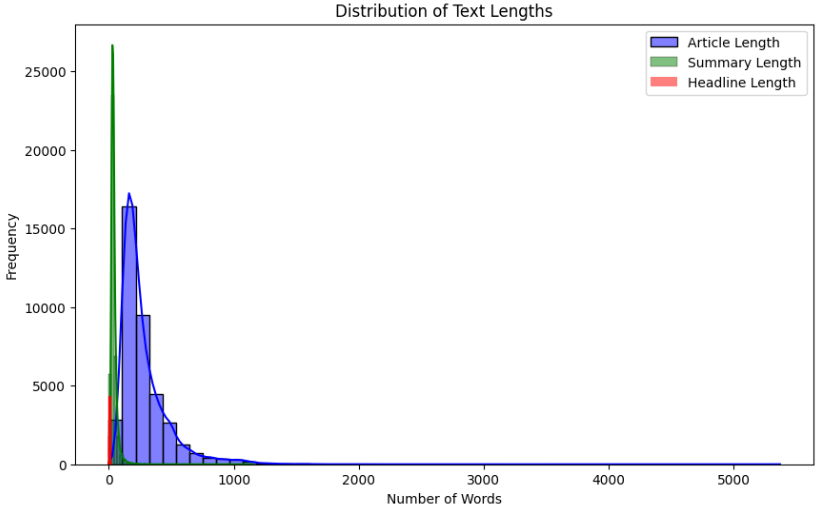


Figure 3.3.5. Distribution of Text Lengths (Headline, Summary, Article)

3.3.6 – Box plot of Article Lengths

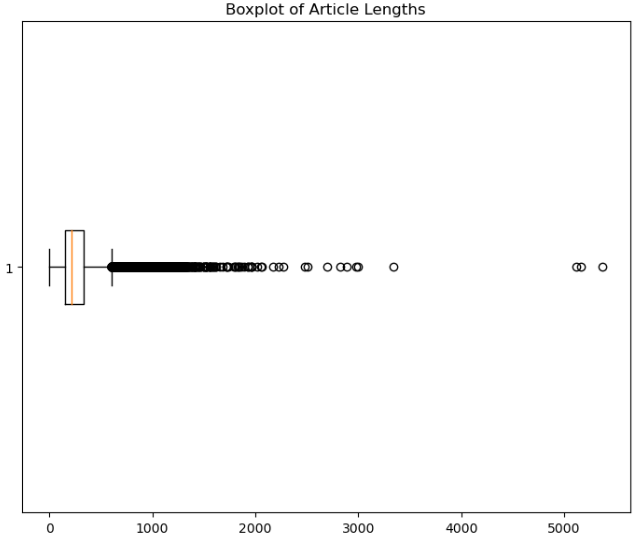


Figure 3.3.6. Box plot of Article Length

3.3.7 – Box plot of Summary Lengths

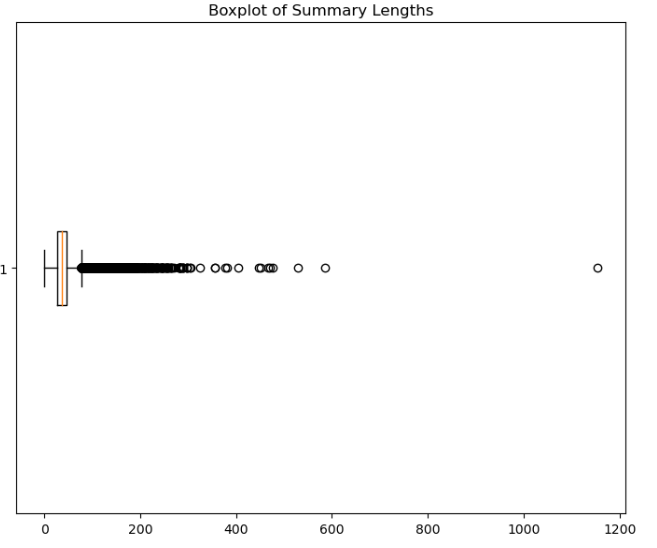


Figure 3.3.7. Box plot of Summary Lengths

## 3.4 Dataset Preprocessing

3.4.1 - Missing Values: the dataset was checked for missing values. If found, the rows were dropped as missing values can affect the accuracy and reliability of machine learning models. If not handled properly, missing data can lead to incorrect predictions, biased results, or errors during training.

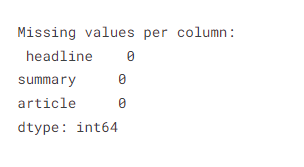
****

Figure 3.4.1. Missing Values

3.4.2- Duplicate Values: the dataset was then checked for duplicate rows of data and was removed to prevent bias outcomes during training and validation. Removing duplicates ensures that the dataset is representative, unbiased, and efficient, leading to more reliable and accurate models.

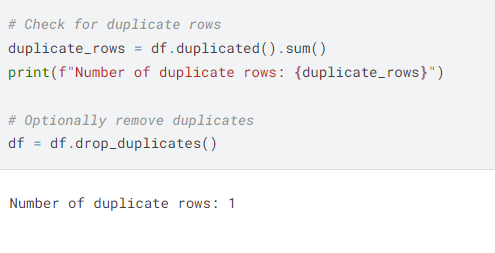


Figure 3.4.2. - Duplicate Values

3.4.3- Outlier Values: article

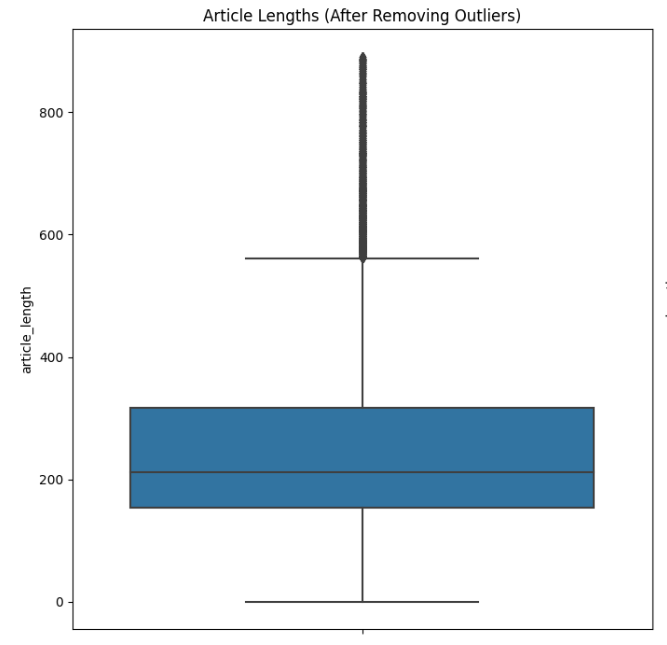


Figure 3.4.3. – Outliers in Article

3.4.4- Outlier Values: summary

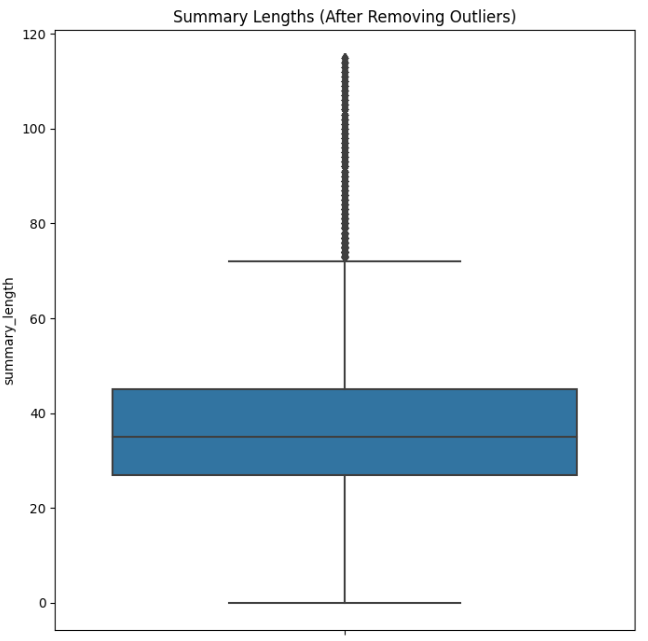


Figure 3.4.4. – Outliers in Summary

## 3.5 Train Test Split

The dataset undergoes an 70-15-15 split, with 70% of the data allocated for training, 15% for validating the summaries generated by our trained model, and the remaining 15% for testing the model’s performance on unseen data. Prior to the split, the dataset is shuffled—a common practice to mitigate any inherent biases stemming from the data's original ordering, where random state = 42.

## 3.6 Models

1) MT5

The mT5 (multilingual T5) model is a multilingual variant of the original T5 (Text-To-Text Transfer Transformer) model which was designed by Google. The mt5 was trained on the mC4 dataset which includes text in 101 languages, which makes it much powerful for several types of multilingual tasks. The languages include Bengali, Chinese, Hindi, Arabic, French, and many more. Thus, it can be applied to multilingual NLP tasks such as summarization, text classification, text generation, translation.

Architecture:

* mT5 is based on the T5 model, which uses the Transformer architecture in an encoder-decoder format. This means that it can be used for a wide range of NLP tasks where the input is first encoded into a latent space and then decoded to produce the output.
* It follows the "Text-to-Text" paradigm, which means that any task such as translation, summarization, classification; is transformed into a text-to-text problem.

Equations:

1. Self-Attention Mechanism: mT5 model uses a self-attention mechanism to weigh the importance of different tokens in the input sequence.

* Q (Query): The matrix representing the input embeddings
* K (Key): The matrix that determines how much focus should be given to other tokens
* V (Value): The matrix containing the values to be used in the output
* **:** The dimensions of the keys used for scaling

1. Masked Language Modeling (MLM): mt5 is trained using a denoising autoencoder approach where some tokens are masked, and the model learns to predict these masked tokens based on the context

* M: Set of masked positions
* : The masked token at position i
* : All other tokens in the sequence

1. Cross-Entropy Loss: The loss for sequence-to-sequence tasks in mt5 is computed using cross-entropy. This calculates the difference between the predicted and actual tokens

* N: No. of sequences
* T: Length of the target sequence
* : The target token at time t
* : input sequence

1. Multi-Head Attention: mt5 employs multi-head attention to capture various types of relationships in the data

* : Learned weight matrices for each head
* : Number of attention heads

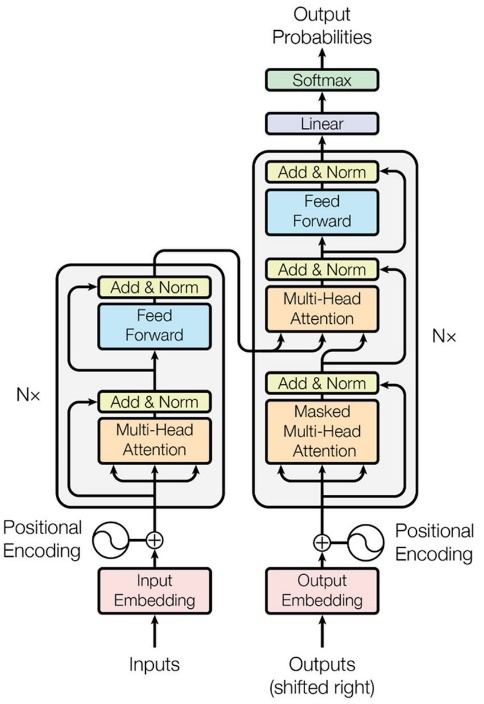
1. Positional Encoding: mT5 uses learnable positional encoding to encode the position of tokens

* : The learned positional embedding for the position pos

1. Feed-Forward Neural Network: Each attention layer is followed by a feed-forward network defined as

* Learnable weight matrices
* : Bias vectors

1. Layer Normalization: After the attention and feed-forward operations, layer normalization is applied to stabilize the training



* : standard deviation of the input vector
* : mean of the input vector
* : learnable parameters

Figure 3.6.1. – MT5 Model Architecture

2) BanglaT5

**BanglaT5** is another version of the T5 (Text-to-Text Transfer Transformer) model fine-tuned specifically for Bengali language tasks. It is also based on the same T5 architecture that treats every NLP task as a text-to-text problem, where both input and output are text sequences. BanglaT5 can also be applied to NLP tasks such as translation, summarization, and question answering in Bengali language. The model is trained on large amounts of Bengali text

Architecture:

Encoder:

* The encoder processes the input sequence by using multiple layers of self-attention and feed-forward networks.
* It generates a representation of the input tokens by attending to all positions in the sequence.

Decoder:

* The decoder generates the target sequence by attending both to its previous tokens and the encoder outputs (via cross-attention).
* It autoregressively predicts the next token in the output sequence, using self-attention to the generated tokens and cross-attention to the encoder outputs.

Equations:

1. Self-Attention Mechanism

* Q (Query): The matrix representing the input embeddings
* K (Key): The matrix that determines how much focus should be given to other tokens
* V (Value): The matrix containing the values to be used in the output
* **:** The dimensions of the keys used for scaling

1. Feed-Forward Neural Network: Each attention layer is followed by a feed-forward network defined as

* Learnable weight matrices
* : Bias vectors

1. Cross-Attention Mechanism:

* comes from decoder, and

3) mbart-50

The mBART (Multilingual BART) model is a sequence-to-sequence model based on the BART architecture, specifically designed for multilingual tasks. It can be fine-tuned for various tasks like translation, text summarization, and many such tasks across multiple languages. In this paper we have used mbart-50 for our summarization task. The mBART-50 is pretrained on denoising autoencoding tasks across 50 different languages, making it effective for translation, summarization, and text generation tasks in many languages.

Architecture:

* It has a bidirectional encoder and an autoregressive decoder:
  + The encoder reads input sequences from all languages and learns bidirectional representations.
  + The decoder generates output sequences autoregressively, token by token, based on both the encoder’s representations and the tokens generated so far.

Equations:

1. Self-Attention Mechanism: mT5 model uses a self-attention mechanism to weigh the importance of different tokens in the input sequence.

* Q (Query): The matrix representing the input embeddings
* K (Key): The matrix that determines how much focus should be given to other tokens
* V (Value): The matrix containing the values to be used in the output
* **:** The dimensions of the keys used for scaling

1. Masked Language Modeling (MLM): mt5 is trained using a denoising autoencoder approach where some tokens are masked, and the model learns to predict these masked tokens based on the context

* M: Set of masked positions
* : The masked token at position i
* : All other tokens in the sequence

1. Cross-Entropy Loss: The loss for sequence-to-sequence tasks in mt5 is computed using cross-entropy. This calculates the difference between the predicted and actual tokens

* N: No. of sequences
* T: Length of the target sequence
* : The target token at time t
* : input sequence

1. Multi-Head Attention: mt5 employs multi-head attention to capture various types of relationships in the data

* : Learnable weight matrices for each head
* : Number of attention heads

1. Positional Encoding: mT5 uses learnable positional encoding to encode the position of tokens

* : The learned positional embedding for the position pos
* i: dimension index
* : The dimension of the models’ embeddings

1. Feed-Forward Neural Network: Each attention layer is followed by a feed-forward network defined as

* Learnable weight matrices
* : Bias vectors

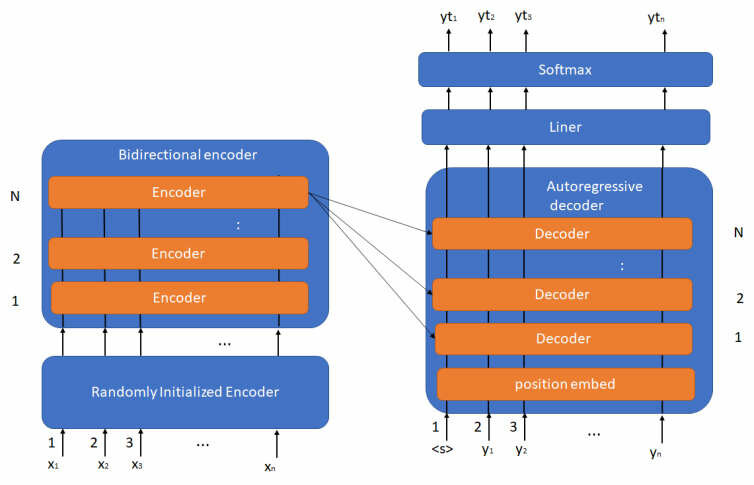


Figure 3.6.3. – mBart-50 Model Architecture

4) IndicBART

IndicBART is a variant of the BART (Bidirectional and Auto-Regressive Transformers) model which was designed for Indic languages; the languages in the Indian subcontinent. As Bengali language is a part of this continent, this model would be well suited for tasks like text generation, text summarization etc.

Architecture:

* Based on the BART model architecture, which combines the strengths of both encoder and decoder architectures.
* Supports both sequence-to-sequence tasks and various NLP tasks, including summarization, translation, and text generation.
* The attention mechanism helps the model focus on relevant parts of the input sequence while generating outputs, improving the quality of the generated text.

The architecture of a transformer consists of an encoder and a decoder. Each layer in both the encoder and decoder has the following components:

Where:

* Each head i is defined as:
* : Learned projection matrices for each head
* Feed-Forward Neural Network (FFN):

Where:

5) Bengali Llama:

Bengali Llama is a specialized adaptation of the Llama (Large Language Model Meta AI) architecture, optimized for the Bengali language. Unlike general-purpose transformer models, Bengali Llama is fine-tuned on a large corpus of Bengali text, making it particularly effective for tasks such as text generation, text summarization, and translation.

Architecture:

* Bengali LLaMA is based on the LLaMA model architecture, which follows a decoder-only transformer design, optimized for autoregressive text generation. This structure allows it to predict the next token in a sequence with high accuracy, ensuring coherence and contextual relevance in generated text.

The core mechanism of Bengali LLaMA relies on the self-attention function:

Where:

* Each head i is defined as:
* : Learned projection matrices for each head
* Feed-Forward Neural Network (FFN):

Where:

## 3.7 Evaluation Metrics

In this study, we assessed the performance of our machine learning models using several key metrics: ROUGE-1, ROUGE-2, ROUGE-L

ROUGE-1: measures the overlap of unigrams (single words) between the generated summary and the reference summaries. It captures the basic content similarity at the word level.

ROUGE-2: measures the overlap of bigrams (pairs of consecutive words) between the generated summary and the reference summaries. It captures some of the structural and contextual information.

ROUGE-L: measures the longest common subsequence (LCS) between the generated summary and the reference summaries. It takes into account the order of words, which allows it to capture some syntactic structure of the sentences.

where S is the generated summary, R is the reference summary, and LCS is Longest Common Subsequence.

## 3.8 Software Implementation

The following software have been used in our Bengali News Summarization Project:

1. **Python**:

The primary programming language for data preprocessing, model training, evaluation, and summarization tasks.

1. **Hugging Face Transformers Library**:

Utilized for loading the pre-trained models used in this project and fine-tuning it for Bengali news summarization and provides functionalities for tokenization, model training, and evaluation. Key components include:

* 1. **Seq2SeqTrainer**: For managing the training process.
  2. **AutoTokenizer**: For efficient tokenization of text.

1. **PyTorch**:

A deep learning framework that serves as the backend for training the transformer model, supporting the implementation of fine-tuning and GPU acceleration.

1. **Pandas**:

Used for data handling and preprocessing tasks, including loading datasets, tokenizing text, and analyzing results.

1. **NumPy**:

Facilitates numerical operations and array handling during data processing and evaluation.

1. **ROUGE**:

An evaluation library for computing ROUGE scores (ROUGE-1, ROUGE-2, ROUGE-L) to assess the quality of generated summaries against reference summaries.

1. **Matplotlib & Seaborn**:

Libraries employed for visualizing training loss, validation loss, and model performance metrics, ensuring a clear presentation of results.

1. **Kaggle Notebooks**:

Interactive environments for writing and executing code, testing models, and visualizing results. Kaggle was utilized as it provides GPU support of up to 16GB.

# Chapter 4 Investigation/Experiment, Result, Analysis and Discussion

## 4.1 Investigation/Experiment

In this chapter, we describe the steps taken during the experimentation phase of the Bengali News Summarization project. We focused on data preprocessing, fine-tuning natural language processing models, and evaluating the summarization system to ensure accuracy and effectiveness in generating concise summaries from Bengali news articles

4.1.1Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | Optimizer | Epoch | Learning Rate | Weight Decay |
| mt5-small | Adam | 10 | 1e-3 | 0.01 |
| mt5-small | Adam | 10 | 1e-3 | 0.01 |
| IndicBART | Adam | 3 | 1e-3 | 0.01 |
| banglaT5 | Adam | 10 | 1e-3 | 0.01 |
| mbart-50 | Adam | 3 | 1e-3 | 0.01 |
| Bengali Llama | Adam | 3 | 1e-3 | 0.01 |

The parameters play an important role in determining the overall performance and behavior of the model. After hyperparameter tuning, the best results were obtained from the following parameters

Table I. Parameters used in different models after hyperparameter tuning

## 4.2 Results

**Results on Existing Dataset (Rouge Scores):**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Rouge -1 | Rouge -2 | Rouge -L |
| mT5-small | 9.33 | 4.05 | 8.41 |
| mT5-base | 1.78 | 0.02 | 1.78 |
| BanglaT5 | 1.41 | 0.51 | 1.40 |
| IndicBART | 28.72 | 8.95 | 16.97 |
| mBart-50 | 23.95 | 9.70 | 16.77 |
| **Bengali Llama** | **67.08** | **49.90** | **33.13** |

Table II. Results of different models on Existing XL-Sum Bengali Dataset

The evaluation of transformer models on the existing XLSum-Bengali dataset shows notable variations in performance. The mT5-small and mT5-base models performed poorly, with ROUGE-1, ROUGE-2, and ROUGE-L scores of 9.33, 4.05, 8.41 and 1.78, 0.02, 1.78, respectively. These results indicate that mT5 models struggle with Bengali summarization, likely due to limited exposure to high-quality Bengali text during pretraining. BanglaT5 showed only a marginal improvement (1.41, 0.51, 1.40), suggesting that it is not well-optimized for this dataset. IndicBART performed significantly better, achieving 28.72, 8.95, and 16.97, indicating its ability to capture meaningful representations of Bengali text. However, mBART-50, which previously performed well on our custom dataset, showed slightly lower scores (23.95, 9.70, 16.77), possibly due to differences in domain-specific text distribution. The most remarkable improvement comes from Bengali Llama, which outperforms all other models with scores of 67.08, 49.90, and 33.13. This significant leap indicates that Bengali Llama is highly effective at summarizing Bengali text, capturing complex linguistic patterns, and maintaining coherence. Compared to IndicBART and mBART-50, it demonstrates a much stronger ability to generate high-quality summaries, making it the most suitable model for Bengali summarization on the XLSum dataset.

**Results on Our Custom Dataset (Rouge Scores):**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Rouge -1 | Rouge -2 | Rouge -L |
| mT5-small | 10.32 | 5.45 | 9.87 |
| mT5-base | 0.78 | 0.04 | 0.78 |
| BanglaT5 | 1.41 | 0.72 | 1.41 |
| IndicBART | 31.97 | 11.36 | 20.69 |
| mBart-50 | 48.56 | 44.44 | 47.40 |
| **Bengali Llama** | **91.62** | **91.23** | **91.62** |

Table III. Results of different models on Custom Scrapped Ittefaq News Dataset

The evaluation of various transformer models on our custom Bengali news summarization dataset, scraped from the Ittefaq news portal, highlights significant differences in performance. The mT5-small and mT5-base models performed poorly, with ROUGE-1, ROUGE-2, and ROUGE-L scores of 10.32, 5.45, 9.87, and 0.78, 0.04, 0.78, respectively. These results indicate that the mT5 models struggle to effectively capture the linguistic and contextual structures of Bengali text. BanglaT5 demonstrated a slight improvement (1.41, 0.72, 1.41), though its overall performance remains inadequate for generating high-quality summaries. IndicBART, benefiting from multilingual training, showed a substantial increase in performance (31.97, 11.36, 20.69), but still lagged behind the strongest models. mBART-50 significantly outperformed all previous models, achieving 48.56, 44.44, and 47.40, suggesting that its extensive multilingual training provides a strong advantage in Bengali summarization. However, Bengali Llama far surpasses all other models, achieving 91.62, 91.23, and 91.62, indicating exceptional performance in capturing Bengali text structures, contextual dependencies, and coherence. The massive increase in ROUGE scores suggests that Bengali Llama is highly optimized for the task, making it the most effective model for summarizing Bengali news articles from our dataset.

## 4.3 Graphs

Training and Validation Losses of Different Models on our Custom Dataset:

1. mBart-50

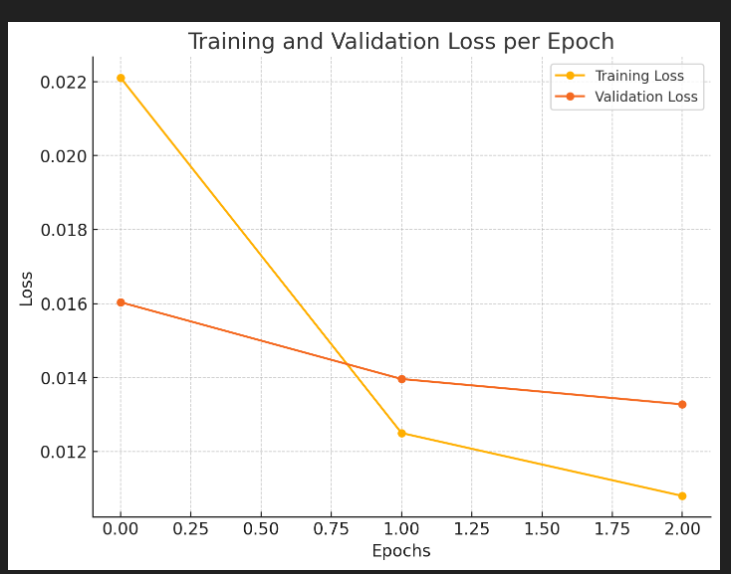


Figure 4.3.1. – Training & Validation Loss of mBart-50

1. BanglaT5:



Figure 4.3.2. – Training & Validation Loss of BanglaT5

1. mT5-base:

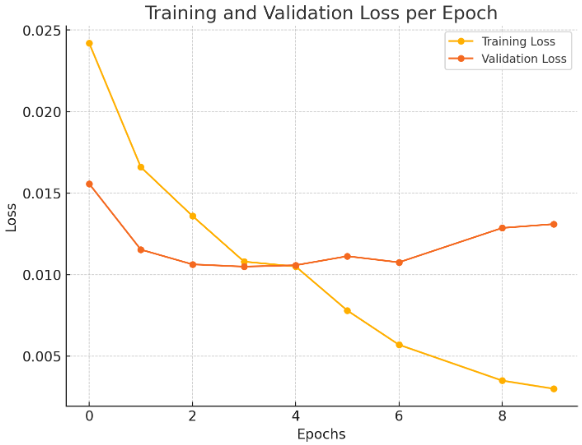


Figure 4.3.3. – Training & Validation Loss of mT5-base

1. mt5-small:

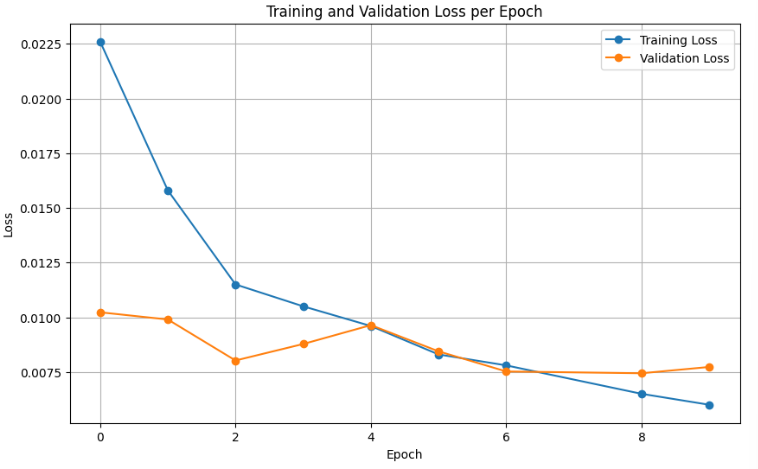


Figure 4.3.4. – Training & Validation Loss of mT5-small

1. Bengali Llama:

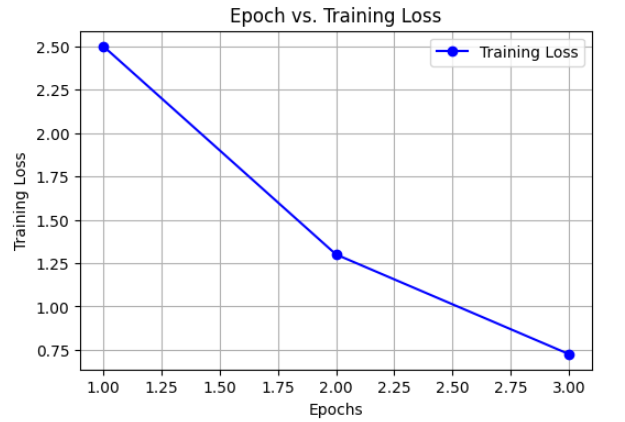


Figure 4.3.5. – Training & Validation Loss of Bengali Llama

## 4.4 Analysis

To assess the quality of the generated summaries, a qualitative analysis was performed. A random selection of generated Bengali summaries was compared to the reference summaries. In general, the model captured the main points of the articles well. However, some generated summaries occasionally missed fine details, especially in longer articles.

## 4.5 Discussion

[Describe the experiments performed addressing all the variables, various results of the project with appropriate figures, tables and texts. The tables and figures should contain appropriate brief captions. Figures should contain appropriate axis labels and legends. The tables and figures should be cited in the project. Perform in-depth analyses of the results represented by each of the figures and tables and finally perform a constructive discussion on the outcome.]

# Chapter 5 Conclusions

## 8.1 Summary

In this project, we explored the use of advanced multilingual transformer models for Bengali news summarization, focusing on models like mT5, IndicBART, mBART-50, BanglaT5, and Ben-Llama. Through an extensive evaluation of these models on various ROUGE metrics (ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-L-Sum), we demonstrated the effectiveness of fine-tuning pre-trained models for generating concise and accurate summaries of Bengali news articles. Our results show that Ben-Llama outperforms other models across all metrics, achieving the highest ROUGE scores, which highlights its superiority for Bengali text generation tasks. The analysis also indicated that the choice of optimizer (AdamW) and learning rate significantly impacts the model’s performance, emphasizing the importance of tuning these hyperparameters during the training process. By leveraging these multilingual pre-trained models, we were able to achieve robust summarization performance, even with the challenge posed by Bengali’s unique linguistic structure. The insights gained from this project lay the foundation for further research and development in the domain of Bengali natural language processing, with applications in media, education, and automated content generation. Moving forward, the next steps could involve exploring additional models, applying more advanced techniques like reinforcement learning, and incorporating domain-specific fine-tuning for even better performance in real-world scenarios.

## 8.2 Limitations

One limitation noted was that the model occasionally missed finer details in longer articles, prioritizing conciseness over completeness. This could be improved by adjusting the model's configuration, such as increasing the maximum sequence length or incorporating post-processing steps to refine the summaries.

## 8.3 Future Improvement

There are more possible future improvements on this project. One significant improvement would be expanding the dataset by collecting a larger and diverse collection of Bengali news articles. This would enable the model to generalize better across various domains such as politics, entertainment, sports, and international news, improving the quality of the generated summaries. Even though ROUGE scores were used to evaluate the model, implementation of other metrics such as BLEU, METEOR, and human evaluations would provide a better assessment of the summary quality. Implementing the model as a real-time service would be another future goal. Integrating it into news websites or mobile apps to automatically generate summaries for live news readers would make it more accessible for users.

# References

Mt5 Link:<https://www.researchgate.net/figure/Model-architecture-of-Transformer-2-that-was-used-in-the-mT5-model_fig1_355850420>

Mbart: Link:<https://www.researchgate.net/figure/The-BART-model-architecture_fig1_364721323>

|  |  |
| --- | --- |
| [1] | N. Moratanch and S. Chitrakala, "A survey on extractive text summarization," *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP),* pp. 1-6, 2017. |
| [2] | N. Ramesh, Z. Bowen, N. d. Cicero, G. Caglar and X. Bing, "Sequence-to-Sequence RNNs for Text Summarization," *CoRR,* vol. abs/1602.06023, 2016. |
| [3] | L. Yang, "Fine-tune {BERT} for Extractive Summarization," *CoRR,* vol. abs/1903.10318, 2019. |
| [4] | R. Tangsali, A. Pingle, A. Vyawahare, I. Joshi and R. Joshi, "Implementing deep learning-based approaches for article summarization in Indian languages," *arXiv preprint arXiv:2212.05702,* 2022. |
| [5] | N. Hayatin, K. Ghufron and G. Wicaksono, "Summarization of COVID-19 news documents deep learning-based using transformer architecture," *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 19, pp. 754-761, 2021. |
|  |  |