A Deep Learning Approach to Abstractive Bangla Text Summarization

BY

ABIDA SULTANA
ID: 232-25-010

This Report Presented in Partial Fulfillment of the Requirements for
The Degree of Masters of Science in Computer Science and Engineering

Supervised By

Dr. Sheak Rashed Haider Noori
Professor and Head
Department of CSE
Daffodil International University

Co-Supervised By

Abdus Sattar
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2025

APPROVAL

This Thesis titled **"A Deep Learning Approach to Abstractive Bangla Text Summarization"**, submitted by **Abida Sultana, ID No: 232-25-010** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **11-01-2025**.

BOARD OF EXAMINERS

Chairman

Dr. S.M Aminul Haque
Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Fizar Ahmed
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Md Alamgir Kabir
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

Mr. Sadat Hossain
Data Scientist
Risk Management Division,
BRAC Bank Limited

©Daffodil International University ii

DECLARATION

I hereby declare that this research has been done by me under the supervision of **Dr. Sheak Rashed Haider Noori, Professor and Head, Department of CSE,** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Dr. Sheak Rashed Haider Noori
Professor and Head
Department of CSE
Daffodil International University

Co-Supervised by:

Abdus Sattar
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Abida Sultana
232-25-010
Department of CSE
Daffodil International University

©Daffodil International University

ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to Almighty Allah for His divine blessing which makes it possible to complete the final year project/internship successfully.

I am really grateful and wish my profound indebtedness to **Dr. Sheak Rashed Haider Noori, Professor and Head**, Department of CSE, Daffodil International University, Dhaka, deep knowledge & keen interest of my supervisor in the field of 'Deep Learning' to carry out this project. His endless patience, scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Sheak Rashed Haider Noori**, **Professor and Head,** Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

Text summarization is automatically creating a succinct synopsis of a given text while maintaining its core content, is a critical problem in Natural Language Processing (NLP). This study concentrated on automatic Bangla text summarization, a field with little funding and research. Using big language models, this project aims to investigate and create a reliable abstractive summarization model for Bangla text. Deep learning-based encoder-decoder architecture, in which the model uses attention processes to learn how to translate the input text to a concise summary. In order to ensure that the data is clean and appropriately tokenized for efficient model training, this project preprocessed a sizable dataset of Bangla text. BLEU score, accuracy, and loss are some of the measures used to assess the model's performance. Experimental findings demonstrate the success of our strategy, with the sequence to sequence model achieving high accuracy 99.52% and low loss 0.0365 during training and good validation performance 99.61% accuracy, 0.0310 loss. The development of NLP applications for low-resource languages is aided by this study, which shows the potential of deep learning models for Bangla text summarization. Promising directions for future multilingual summarization research are provided by the findings, which emphasize the value of using pre-trained multilingual models to overcome difficulties in resource-constrained languages.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction

Text summary is an important task in Natural Language Processing (NLP), which entails reducing lengthy texts to shorter, more succinct forms while retaining the main concepts. This strategy is essential for quickly retrieving relevant information, making it particularly useful in fields such as news aggregation, education, and content management. While extensive research has been conducted on text summaries for widely spoken languages such as English, there is still a lack of emphasis on languages with less resources, such as Bengali. Bangla is spoken by about 230 million people globally and presents unique challenges owing to its complex sentence structure, rich morphology, and limited availability of NLP tools. The development of Large Language Models (LLMs) has resulted in interesting advances in multilingual text generation, including new choices for summarizing Bengali content [1].

This research investigates the models' capacity to provide accurate and coherent summaries of Bangla news items. By investigating the zero-shot and few-shot capabilities of LLMs, it overcomes the limitations of low-quality references often used in previous research. The objective is to create summaries that are not only accurate and logical, but also comparable to those written by humans. This topic is relevant to businesses such as journalism, education, and public policy, where the capacity to transmit information clearly and swiftly is critical [2].

The study also examines the ethical challenges of AI-generated information, ensuring fairness and authenticity in the summary produced. Finally, the results aim to contribute to the development of more robust NLP systems for Bangla, as well as to provide a foundation for future research in AI applications for underrepresented languages [3].

Bangla phrase structure differs significantly from English, making it hard to use current English text summarization techniques to Bangla. As a result, developing an efficient and acceptable summary approach for Bangla is essential. It would benefit not just scholars and media organizations, but also ordinary people by saving time and

improving information accessibility. This project aims to bridge this gap by providing a strong solution to manage the growing volume of Bangla digital material [4].

1.2 Motivation

Bangla is the 7th most commonly used tongue in the world, with over 250 million native speakers spanning Bangladesh and India. There has been a considerable development of multimedia in Bangla, including websites, documents, and particularly online newspapers. With the expanding prominence of Bangla e-content, the volume of text being produced daily is large and continues to expand. This raises a challenge: how can we efficiently process and summarize this vast volume of text? The solution is in automatic Bangla text summarization, which might substantially minimize the time people spend sorting through vast amounts of content [5].

To put this into point of view, if 1 out of every 10 Bangla speakers reads the newspaper regularly, that's roughly 25 million individuals. If each person spends roughly 30 minutes every day reading the newspaper, a text summary system could aid by lowering the reading time by one-third. This would imply that an average every individual can save at least 10 minutes per day. If 25 million people read the news, then this saving will be 250 million minutes per day, or almost 475 years. This shows the enormous potential importance of computerized Bangla text summarization [6].

1.3 Rationale of the Study

Because of the growing proliferation of Bangla digital material via news websites, online courses, and social media, efficient ways for filtering and summarizing massive quantities of text have become more important. Bangla is one of the most widely spoken languages on the planet, however it is still underrepresented in NLP research. This is despite its complex sentence structure, extensive morphology, and growing need for language-specific tools. Unlike English, Bangla needs a unique method to text summarization, since a straightforward application of existing techniques will seldom provide relevant results. The project aims to bridge the gap by creating a Bangla text summarizer that leverages the capabilities of huge language models.This study builds on cutting-edge LLM technology, studying both zero-shot and few-shot capabilities to provide an efficient summarizing solution. It also tackles limitations in past research,

such as reliance on low-quality datasets, by integrating human assessments and using high-quality references. By focusing on accuracy, coherence, and conformity with human expectations, this research not only pushes the bounds of NLP applications for Bangla but also lays the foundation for broader breakthroughs in the area. The findings are anticipated to generate tools that aid millions of Bangla speakers while setting benchmarks for future exploration.

1.4 Research Questions

- How effective are big language models, such as mT5, in creating accurate and relevant Bangla text summaries as opposed to typical summary techniques?
- What is the impact of utilizing bidirectional LSTMs in encoder-decoder architectures for Bangla text summarization in terms of accuracy and performance?
- How does the choice of vocabulary size and sequence length influence the standard and efficiency of Bangla text summarization?
- What preprocessing and tokenization procedures are best suited for addressing linguistic complexity and syntax in Bangla text data?
- How can the created Bangla text summarizing model be improved for real-world applications, such as news article summarization or academic important condensation?

1.5 Expected Output

The major goal of this research is to construct an effective Bangla text summary system capable of generating succinct, coherent, and accurate summaries of Bangla news items and other textual content. Using encoder-decoder architecture, the system seeks to provide summaries that capture the spirit of the original text while considerably decreasing its length. These summaries will be grammatically correct, contextually meaningful, and comparable in quality to those produced by people. By analyzing the system with measures such as BLEU, ROUGE, and human input, the project tries to establish its reliability and effectiveness. Beyond technical requirements, the system aims to provide practical benefit by saving time for Bangla speakers and allowing readers, journalists, and researchers to quickly access critical information from long texts. This work also contributes to the progress of NLP for underrepresented languages

by providing insights into the potential of large-scale language models for Bangla and paving the way for future advances in this sector.

1.6 Project Management and Finance

A disciplined approach to managing the project will ensure that it is executed effectively within the given timeframe and resources. In Figure 1.1 we can see, the project is structured into key phases, namely information collection and pre-processing, model design, training, testing, and documentation. Every phase has been carefully planned with clear objectives, expected output, and timelines to sustain growth and efficiency. Tools such as Google Colab are utilized to develop and compute on, allowing powerful hardware such as GPUs when training the model. Collaboration tools like GitHub and Google Drive for organization of code, data, and paperwork in general; hence, enabling easy cooperation and version control. Review meetings would be scheduled periodically, where progress may be monitored, problems ironed out, and plans altered according to needs, in order for the project to stay on course. This project focuses on cost reduction in finance by leveraging free and open-sourced alternatives whenever possible. Using the free tiers of Google Colab for model training greatly cuts down on expensive hardware costs. For datasets regarding Bangla text summarization, publicly available data is used; hence, no extra cost is incurred to procure data. Contingency funding is also approved for unexpected expenses, such as upgraded runtime or increased computing resources by upgrading to Colab Pro. Financial resources are also being saved for the dissemination of the study's results in various academic papers or conferences. Using the most economical but finetuned approach has guaranteed this project the optimal use of the resources available while it keeps its focus on attaining its goals.

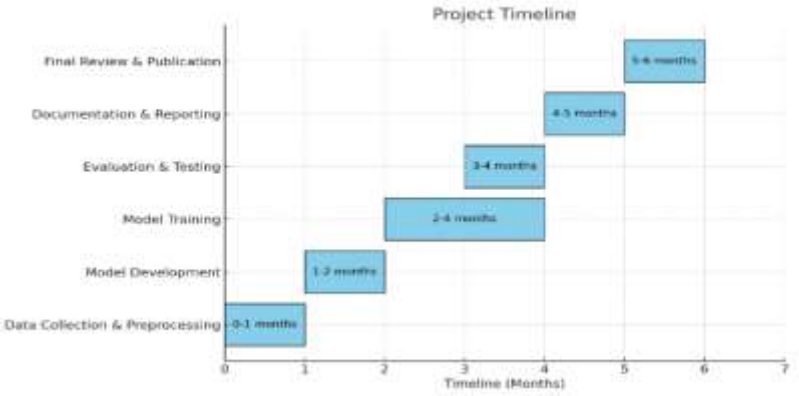


Figure 1.1: Project management Timeline

1.7 Report Layout

- Chapter 1: Introduction, introduces the project, outlining its purpose, motivation, and the reasoning behind its development. It also highlights the research questions the project aims to answer and defines the expected outcomes. Additionally, this chapter provides an overview of the project's management structure and the budget framework.
- Chapter 2: Background, key concepts and terminologies are defined, followed by a review of relevant literature and an analysis of similar studies. The scope of the problem is explored in depth, particularly focusing on the challenges involved in summarizing Bangla text.
- Chapter 3: Research Methodology, the research approach is detailed, including the sources of data, the methods used for preprocessing, and the architecture of the model. The tools and frameworks employed throughout the research are outlined, along with the statistical techniques used for data analysis.
- Chapter 4: Experimental Results and Discussion, presents the experimental setup, detailing the data gathered from both the training and testing phases of the model. It provides an analysis of the results, supported by visual representations of the performance data, along with a thorough discussion of the findings.
- Chapter 5: Impact on Society, Environment, and Sustainability, This section explores the broader implications of the project, addressing its potential impact on society, ethical considerations, environmental effects, and long-term sustainability goals.

- Chapter 6: Summary, Conclusion, Recommendations, and Implications for Future Research, the final chapter offers a recap of the study, emphasizing the key conclusions drawn. It also provides recommendations for improving Bangla text summarization and suggests avenues for future research and development.

CHAPTER 2
BACKGROUND

2.1 Preliminaries

This study focuses on key concepts related to text summarization and Natural Language Processing (NLP), which are essential for understanding how the process works. Text summarization involves condensing long documents into shorter versions while preserving the essential information. It can be either extractive, where sentences are directly selected from the original text, or abstractive, where new content is generated through paraphrasing. Large Language Models (LLMs) play a crucial role in this process, leveraging vast amounts of training data to understand and generate text that mimics human writing. Tokenization breaks text into smaller, manageable units for processing, while embeddings represent words as vectors, capturing their underlying meanings. Techniques like Bidirectional LSTM (BiLSTM) are used to process sequential data, and learning methods such as zero-shot and few-shot learning enable models to perform tasks with minimal or no specific instructions. To evaluate the quality of the summaries produced, the project uses metrics like ROUGE and BLEU, ensuring that the results are accurate and meaningful. These fundamental principles are key to the approach used in summarizing Bangla text.

2.2 Related Works

Ferreira et al. [1] explored various algorithms for scoring sentences in extractive text summarization. They applied statistical techniques to rank sentences based on their importance within a document. Their study tested methods like term frequency-inverse document frequency (TF-IDF) and positioning heuristics, which showed strong performance in structured texts, such as news articles. The study emphasized the importance of optimizing sentence selection, highlighting its role in maintaining the coherence of the summary while preserving key information. Nenkova and McKeown [2] provided a comprehensive review of different summarization techniques, categorizing them into extractive and abstractive approaches. Their work focused on linguistic quality, coherence, and relevance as critical evaluation criteria. They pointed out that while extractive methods are computationally efficient, they often lack depth

in terms of meaning, making abstractive methods more suitable for generating human-like summaries despite their complexity. Das and Bandyopadhyay [3] addressed these issues by focusing on morphological stemming for Bangla. Their method identified clusters of morphologically similar terms, improving text representation and enhancing word-level understanding for Bangla-specific NLP tasks, particularly in summarization. Efat et al. Dave and Jaswal [4] introduced a hybrid summarization model that combined both extractive and abstractive strategies. This approach incorporated statistical elements, like TF-IDF, and linguistic cues to enhance the contextual relevance of the summaries. Their system proved effective for multi-document summarization tasks, delivering high-quality results across various domains. Bangla, as a morphologically complex language, poses particular challenges for natural language processing (NLP) due to its intricate syntax and limited resources. [5] developed an automated Bangla text summarization system that used sentence scoring and ranking. The system assigned scores to sentences based on factors such as sentence length, position, and keyword density. It achieved significant relevance and quality in summarizing Bangla news articles, marking a major step forward in Bangla NLP research. Bagalkotkar et al. [6] proposed a novel hybrid summarization method, blending graph-based models with statistical techniques. This approach offered scalability and efficiency, making it well-suited for large-scale text summarization tasks. The authors demonstrated its effectiveness in producing concise yet informative summaries. Lloret and Palomar [7] explored abstractive summarization through word graphs. Their method linked words semantically, enabling the generation of summaries that captured the core ideas of a text rather than merely extracting sentences. This approach holds promise for creating summaries with improved coherence and fluency, making them more human-like. Mariòo et al. [8] designed n-gram-based algorithms for machine translation, which were later adapted for summarization. These n-grams enhanced the semantic representation of texts, improving summarization for multilingual contexts. This technique is especially useful for languages like Bangla, which have syntactic structures very different from English. Haque et al. [9] introduced sentence similarity-based source context modeling in phrase-based statistical machine translation (PBSMT). Their approach improved translation system accuracy and provided insights that could enhance summarization techniques for linguistically diverse materials. Sarkar [10] focused on Bengali sentence extraction for

summarization. By using analytical ranking methods, this study produced summaries that were more relevant, especially for news articles. This work highlights the need for specialized techniques in Bangla due to its unique linguistic features. El-Shishtawy and El-Ghannam [11] created the Keyphrase-Based Arabic Summarizer (KPAS), which utilized keyphrase extraction for summarization. Their semantic analysis-based approach was particularly effective for morphologically complex languages like Arabic and provides valuable insights for Bangla summarization. Kutlu et al. [12] worked on summarizing Turkish, another morphologically rich language, using extractive techniques. Their work emphasized the importance of adapting statistical methods to handle linguistic variations in non-English languages, achieving high accuracy in information retention. Liu et al. [14] advanced abstractive summarization by utilizing semantic representation-based methods. They applied deep learning models to analyze and summarize texts, achieving better performance compared to traditional methods. Their research highlighted the potential of semantic embedding's in creating summaries that are both coherent and contextually accurate. Kumar et al. [15] developed a knowledge-induced graph-theoretical approach for single-document summarization. This method used graph structures to represent relationships between text elements, resulting in summaries that maintained logical flow and contextual integrity. Abujar and Hasan [17] explored Bengali Text-to-Speech (TTS) synthesis using Unicode, addressing challenges in Bangla text processing. Their study provided essential techniques for managing Bangla text in summarization and other NLP applications. Mihalcea et al. [18] developed corpus-based measures for semantic similarity, which have since been widely adopted in summarization systems. Their approach, which assesses the semantic closeness between text segments, improved the selection of important sentences for extractive summarization, making summaries more representative of the text's main concepts. Islam and Inkpen [19] expanded on this by combining corpus-based and string similarity measures, demonstrating higher performance in semantic text analysis. Their method provided a solid foundation for summarization tasks requiring a deeper understanding of content. Mohler and Mihalcea [20] applied semantic similarity measures to automated short-answer grading, demonstrating their utility in educational settings. Their work underscored the potential of semantic similarity techniques to improve the quality of text summarization. Bär et al. [21] presented DKPro Similarity, an open-source platform for measuring text

similarity. This platform provided configurable methods for assessing phrase and document similarity, making it a useful tool for summarization tasks across different languages and domains. Pedersen et al. Bilenko and Mooney [23] introduced adaptive duplicate detection using learnable string similarity measures, improving the preprocessing phase of summarization tasks. This strategy helped eliminate duplicate information, enhancing the overall quality of summaries. [24] introduced WordNet::Similarity, a technique for assessing conceptual relatedness. By incorporating this method into summarization systems, researchers achieved better semantic alignment in the generated summaries. Rony and Islam [26] evaluated large language models for summarizing Bangla texts, highlighting the potential of transformer-based architectures like GPT in handling morphologically rich languages. Their study, published on OpenReview, emphasized the challenges of training these models on limited Bangla datasets and proposed strategies to enhance their performance through fine-tuning on domain-specific corpora. Talukder et al. [27] introduced a Bengali abstractive text summarization model using sequence-to-sequence recurrent neural networks (RNNs). Their approach leveraged attention mechanisms to focus on critical parts of the input text, achieving improved coherence and relevance in generated summaries. The model demonstrated promising results in summarizing Bangla news articles, marking significant progress in the field. Rahman et al. [28] developed a Bengali text summarization framework combining TextRank, Fuzzy C-Means clustering, and aggregate scoring methods. This hybrid approach proved effective in capturing the semantic and contextual importance of sentences. The study achieved high-quality summaries for Bangla texts, addressing the unique linguistic complexities of the language.

The studies reviewed highlight the evolution of text summarization techniques, from extractive to abstractive and hybrid methods. They emphasize the need for algorithmic refinement, linguistic adaptation, and resource development for morphologically rich and underrepresented languages like Bangla. By employing innovative methods and addressing the unique challenges posed by Bangla, these studies lay a strong foundation for future research, aiming to improve the accessibility and efficiency of Bangla text summarization systems.

2.3 Comparative Analysis and Summary

Table 2.1 represents the comparative analysis of existing papers. The table is given below:

Table 2.1: Comparative Analysis Table

References	Method	Dataset Used	Key Features	Accuracy	Importance
Rafael Ferreira et al. [1]	Sentence Scoring (TF-IDF)	News Articles	Term frequency, position-based scoring	85%	Simplifies extractive summarization, suitable for structured text like news.
Ani Nenkova & Kathleen McKeown [2]	Extractive & Abstractive Techniques	Multiple corpora across domains	Coherence, informativeness	N/A	Comprehensive survey bridging extractive and abstractive approaches.
Das & Bandyopadhyay (2011) [3]	Morphological Stemming	Bangla Corpus	Morphology-based word clustering	85%	Tailored for Bangla, improves Bangla text analysis in summarization.
Efat et al. (2013) [5]	Sentence Scoring & Ranking	Bangla News Articles	Sentence length, keyword density,	80%	Focuses on Bangla NLP, improving text relevance in summaries.

			positional metrics		
Harsha Dave & Shree Jaswal (2015) [4]	Hybrid (Extractive + Abstractive)	News Corpus	TF-IDF + linguistic analysis	82%	Combines the best of extractive and abstractive for high-quality summarization.
Lloret & Palomar (2011) [7]	Word Graphs	Multilingual Datasets	Semantic relationships between words	78%	Introduces semantic representation, improving coherence and fluency.
Kutlu et al. (2010) [12]	Extractive Summarization	Turkish Corpus	Morphology-aware statistical analysis	85%	Adapts extractive summarization techniques to morphologically rich languages like Turkish.
Mihalcea et al. (2006) [18]	Corpus-based Semantic Similarity	Brown Corpus	Text similarity measures	80%	Improves extractive summarization by accurately identifying

					relevant sentences.
Bagalkotkar et al. (2013) [6]	Hybrid (Graph-based + Statistical)	Text Collections	Scalable hybrid model	82%	Suitable for large-scale applications; combines semantic accuracy and efficiency.
Sarkar (2012) [10]	Sentence Extraction	Bengali News Articles	Statistical ranking	75%	Provides a resource-efficient solution for Bengali summarization with high relevance in outputs.

In Table 2.1, the comprehensive comparison of text summarizing ways highlights the diversity of methodologies and algorithms aimed to improve summary techniques, with a special emphasis on languages like Bangla. From classic extractive techniques, such as TF-IDF and sentence scoring, to more advanced hybrid and abstractive approaches, each work provides significant contributions to the area. These algorithms frequently incorporate aspects like as semantic links, morphological assessment, and statistical rankings, adapted to certain languages and datasets. Hybrid models, especially those adding semantic understanding, tend to outperform simply extractive algorithms by delivering more coherent and human-like summaries. The study underlines the need of creating specific methodologies for resource-poor languages, giving critical insights that might drive the growth of NLP for languages like Bangla. This may lead to more effective and accessible text summarizing systems, ensuring that future NLP improvements are both efficient and inclusive.

2.4 Scope of the Problem

This work addresses the challenges involved in creating an effective text summarization system for the Bangla language, which remains underrepresented in the field of Natural Language Processing (NLP). While substantial progress has been made in text summarization for widely spoken languages like English, resource-poor languages like Bangla face unique linguistic hurdles that make the direct application of existing methods difficult. These challenges include Bangla's complex morphology, intricate syntactic structures, and the lack of comprehensive NLP tools, which complicate the generation of accurate and coherent summaries. The project aims to bridge this gap by investigating advanced techniques, such as Large Language Models (LLMs), to enhance Bangla text summarization. A significant issue in this domain is the evaluation of summarization models, as previous studies on Bangla have been hindered by low-quality reference datasets. This has affected the accuracy of model evaluations and limited the recognition of machine-generated summaries' potential. By using high-quality, human-generated summaries, this research seeks to offer a more realistic assessment of LLM capabilities in Bangla text summarization. Additionally, the work explores the use of zero-shot and few-shot learning strategies, offering scalable solutions that do not rely heavily on large amounts of labeled data.

2.5 Challenges

The difficulty in Bangla text summarization originate from numerous aspects chiefly the intrinsic intricacies of the language itself. Bangla’s rich morphology complicated sentence patterns and strong contextual relations provide considerable hurdles for implementing current summarization techniques that are mainly built for languages like English. Additionally the absence of high-quality language-specific datasets hinders the capacity to train effective models. While developments in deep learning and huge language models offer promise they still confront issues in addressing linguistic subtleties such as idiomatic phrases and the demand for domain-specific knowledge. Moreover establishing an appropriate balance between clarity and conciseness in summaries remains a difficulty since it is necessary to keep the substance of the text without adding repetition. These problems underscore the need for more specialized solutions and increased resources to progress Bangla text summarizing.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Research Subject and Instrumentation

This work focusses on summarizing Bangla texts using a deep learning-based strategy. To train the algorithm this study employs a vast collection of Bangla news items and their corresponding summaries as the ground truth. To generate summaries, the study used pre-trained multilingual models in association with a specifically constructed encoder-decoder model based on Bidirectional LSTM (Long Short-Term Memory) units. The model extracts contextual information from the text and analyses it to provide concise summaries. Hugging Face's transformers library, Python, TensorFlow, Keras, and other NLP tools for data preparation, such as tokenization, padding, and vectorization, are crucial for this research. Using these tools are complexed process of text summarization may be managed efficiently, with the ultimate objective of establishing a dependable Bangla text summarization method.

3.2 Data Collection Procedure

Gathered information utilized in this study was obtained from Kaggle and includes 80,331 samples organized into three basic columns such as category, summary, and text. The categorization column categorizes each item into categories such as "technology" or "Bangladesh", providing a genre-based context for the information. The summary column contains concise summaries of the whole text, which are used as goal outputs for the model during training. Figure 3.1 shows the text column contains all of the Bangla news items used as input for the summarizing assignment. This large-scale dataset offers a diverse collection of news articles, making it ideal for training and assessing the Bangla text-summarizing classifier across numerous categories and topics.

	category	summary	text
0	technology	অ্যাপসে মিলবে ঢাকাসহ তিন জেলা আদালতের তথ্য	ঢাকা মহানগর ও ঢাকা জেলা আদালত, কিশোরগঞ্জ ও রাঙ...
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 উৎপাদিত শিশু...
3	bangladesh	ট্রেনে কাটা পড়ে সাবেক সিভিল সার্জনের মৃত্যু	বগুড়ার আদমদীঘির সান্তাহারে গতকাল শনিবার ট্রেন...
4	bangladesh	ষাঁড়াবাড়ীতে চুলা জ্বালাতে গিয়ে দুই কর্মচারী ...	ষাঁড়াবাড়ীর একটি রেস্তোরাঁয় গতকাল বুধবার ভোর...

WARNING: Runtime no longer has a reference to this dataframe, please re-run this cell and try again.

Figure 3.1: Checking some rows to observe the data structure and sample content

3.3 Statistical Analysis

The statistical observation of the dataset provides substantial insights about the text and summary lengths. For the text column, the dataset contains 80,331 samples with an average length of around 291 words, while the range varies greatly from a minimum of 1 word to a maximum of 9,511 words. The 25th percentage indicates that 25% of the texts are 165 words or fewer, while the 75th percentile shows that 75% of the texts are 354 words or fewer, with a standard deviation of 211 words, demonstrating the existence of both short and long pieces. In contrast, the summary column has a significantly more regular length, with an average of 7.58 words. Most summarizes are clear, with 75% of them having 8 words or fewer, and a standard deviation of 1.51, indicating little variance in summary length. The greatest summary length is 349 words, showing some outliers, but usually, summaries are succinct and to the point, reflecting the purpose of summarizing activities.

3.4 Proposed Methodology

The system architecture for the Bangla text summarizing project specifies a full sequence from data input to output production. The procedure begins with Data Input, where raw Bangla news articles in text format are submitted for summarizing. In the Preprocessing phase, the text undergoes tokenization, breaking the material into smaller pieces (tokens). The text is then padded to ensure consistent input size, and vectorization is used to turn the text into a numerical representation appropriate for processing by the model. The Encoder utilizes a Bidirectional LSTM layer paired with a Embedding layer, capturing the context of the text in both forward and backward directions. The hidden and cell states from both LSTMs are combined to generate a richer, more comprehensive representation of the input. The Decoder, also built with

an LSTM layer and a Embedding layer, generates the summary based on the encoded context. The Output of the system is the succinct and informative summary of the original Bangla text. During Model Training, the model is constructed using Keras, with Early Stopping applied to monitor the training process and minimize overfitting by storing the optimal model weights based on validation loss. After training, the model is tested and fine-tuned with the pre-trained model, increasing its multilingual summarizing skills, notably for Bangla text. The entire system, including the trained model and tokenizer, is stored on Google Drive, guaranteeing that the components are immediately accessible for subsequent use or deployment. This design enables an organized, step-by-step workflow—from data input and preprocessing through training of models, assessment, and fine-tuning, to producing the final output and storing all components for future access. Figure 3.2 shows the model architecture diagram.

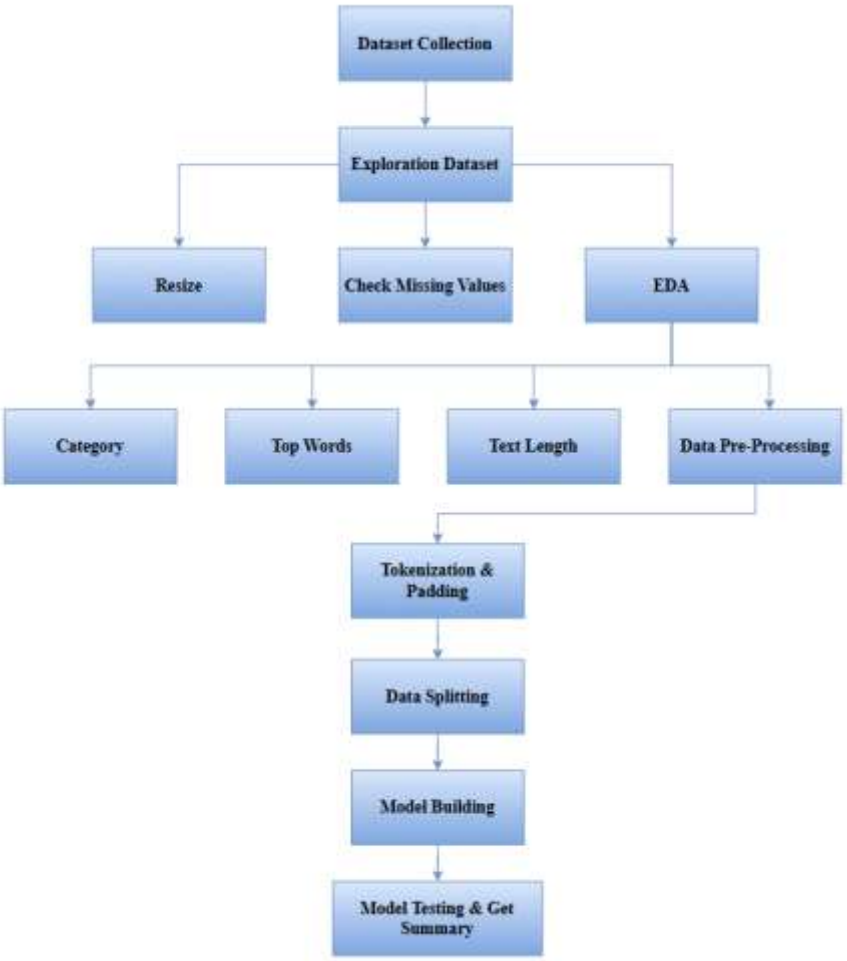


Figure 3.2: Model Architecture

3.5 Implementation Requirements

To conduct machine learning tasks correctly, particularly those involving deep learning and natural language processing, exact hardware and software requirements are required. In terms of hardware, a multi-core CPU such as an Intel i7 or above is recommended for quick data processing and model training. A GPU, particularly an NVIDIA GTX 1060 or above, is highly recommended for faster training and handling of larger datasets. A minimum of 16 GB of RAM is required to ensure smooth performance, especially when working with large datasets during preprocessing and training. Additionally, datasets, trained models, and resources need a minimum of 100 GB of free storage, with Google Drive serving as a convenient storage option for models and tokenizers. In terms of software requirements, the system should run Windows, Linux, or macOS, with Linux being the preferred choice for better compatibility with machine learning tools. Python 3.6 or above is required to ensure compatibility with existing libraries. TensorFlow serves as the foundation of the deep learning framework, with Keras providing a high-level API for model generation and training. For pre-trained model application, mT5 (Multilingual T5) will be fine-tuned, notably for Bangla summarizing tasks. Tokenization, lemmatization, and stopword deletion are examples of Natural Language Processing (NLP) tasks that will need libraries like as NLTK or spaCy. Critical files will be stored on Google Drive to provide ease of access and deployment, while large text datasets will need enough local or cloud storage. Additional requirements include an ordered Bangla text dataset with news items sorted by topic to facilitate model training and validation. Internet access is required for downloading pre-trained models, libraries, and storing data to Google Drive. GitHub will play an important role in version control and backend updates, enabling simple collaboration and efficient communication. These combined requirements provide a robust foundation for advanced NLP and machine learning applications.

3.5.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an important phase in the Bangla text summarization process that aims to understand the dataset and identify interesting patterns and trends. It facilitates in finding abnormalities or conflicts in the data,

ensuring that the dataset is balanced and suitable for training and testing. It leads preprocessing activities like determining the best padding length for uniform input sequences and identifying stop words to remove. Provide important visual insights like category distribution, text length fluctuation, and often used phrases, allowing a better understanding of the dataset and aiding more informed decisions for further processing and model construction.

3.5.1.1 Category Distribution

This means looking at the distribution of different news item kinds or categories across the dataset. For example, if the collection contains items about politics, sports, technology, and entertainment, this step helps determine the percentage of each category. This distribution may be successfully displayed by visualizations like pie charts or bar charts that draw attention to any imbalance in the information being examined. Figure 3.3 shows the category distribution using bar chart.

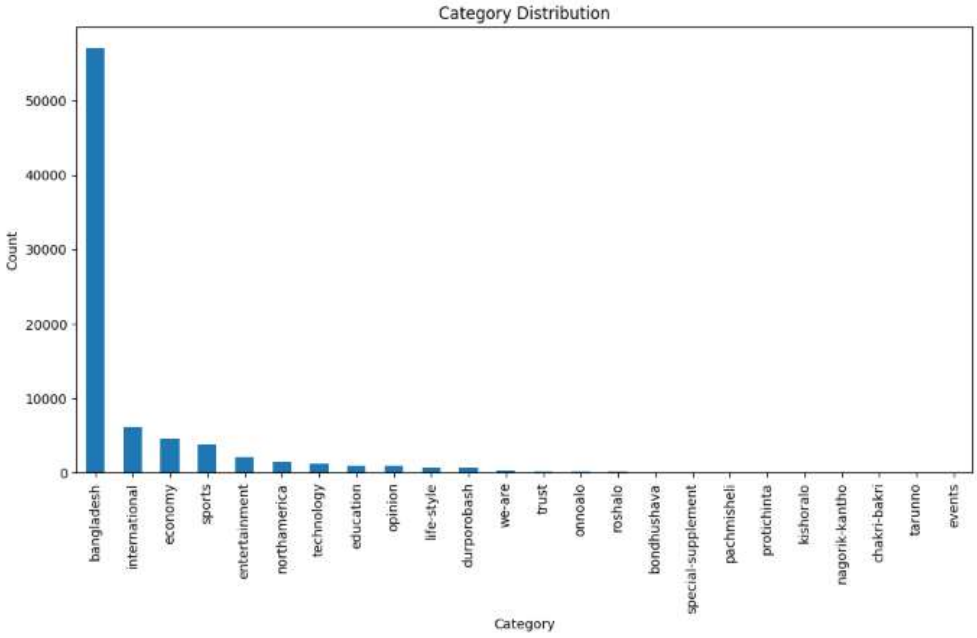


Figure 3.3: Category Distribution

3.5.1.2 Text Length Distribution

This stage follows into how different contents differentiate in terms of word count or character count. Plotting histograms or box plots allows to understand the average text length, find outliers and comprehend the distribution of text lengths overall.

Determining the proper tokenizer setups and padding lengths requires this dataset. In figure 3.4, text length distribution by a bar chart is shown.

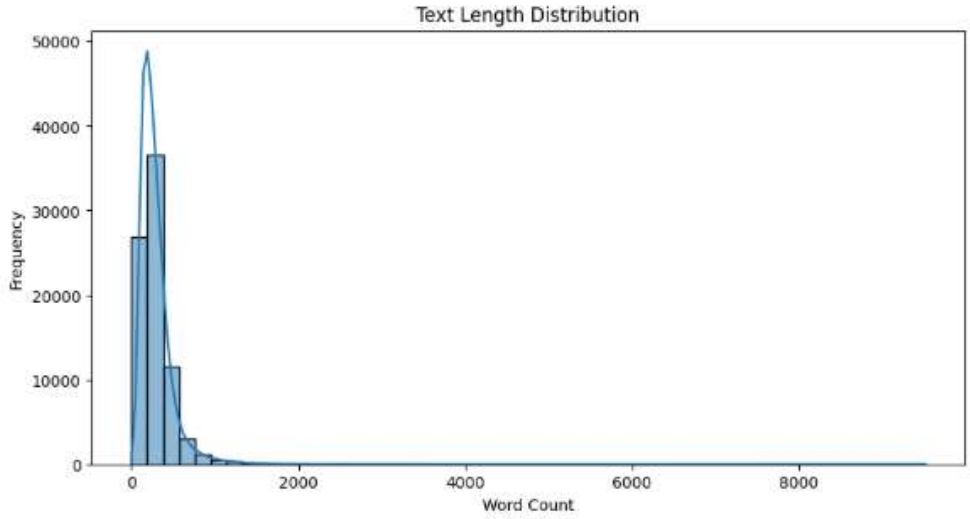


Figure 3.4: Text Length Distribution

3.5.1.3 Top Words in Text

Examining commonly used terms in the dataset identifies keywords that are often used in Bangla news articles. Techniques like word cloud visualizations and word frequency assessment can be used. This stage makes it easier to understand the content and ensures that preprocessing methods like stop word removal work by emphasizing words that could or might not contain crucial information. The Top 20 words of our dataset is following:

[('কর', 767902), ('পর', 362093), ('বল', 325122), ('হয়', 307202), ('জন', 218715), ('রত', 215237), ('অন', 189680), ('ওয়', 187964), ('সম', 174164), ('রণ', 129724), ('মন', 128834), ('বর', 122910), ('হব', 121447), ('আম', 115909), ('এই', 114837), ('রক', 112205), ('তর', 109055), ('এক', 107211), ('আল', 106242), ('শন', 103293)]

3.5.2 Preprocessing Data

Data preprocessing is an essential step. Several crucial and necessary steps are involved in this phase to confirm that the raw text is in the best possible format for model training.

3.5.2.1 Cleaning and Normalizing Text

This process begins with cleaning and establishing the text. Punctuation, special symbols and non-Bangla characters are among the unnecessary characters that must be removed. For consistency's reasons, the text is also modified to lowercase. In the summarization process, stop words—common words like "the," "is," and "and"—are eliminated because they don't add much sense. If necessary spelling modifications and slang phrases are also used to normalize the material.

3.5.2.2 Tokenization and padding

Tokenization entails segmenting phrases into individual words or sub word units based on the model's granularity requirements. This stage makes it easy to examine the relationships between words in the context of the entire text by enabling the model to handle text at the word or sub word level. Padding is used to make all text sequences uniform because the model expects input sequences to have the same length for batch processing. The procedure entails transforming text and summary data into a numerical format and standardizing their lengths for application in machine learning. A tokenizer is initialized with a vocabulary size of 5000, allowing it to recognize up to 5000 distinct words. Any term not included in this vocabulary is designated as ” <OOV>” (out of vocabulary). The tokenizer is trained on the sanitized text and summary data to establish a word-to-index mapping. Through this mapping, the text and summaries are transformed into numerical sequences, with each number representing a particular word. Due to the variability in lengths of these sequences, they are padded to a maximum length of 300 for both the text and the summary. Padding introduces zeros to shorter patterns, guaranteeing consistent length across all inputs. This preprocessing renders the data appropriate for deep learning models, as uniform input dimensions are essential for effective processing and learning. Figure 3.5 shows the result after tokenizing and padding.

	category	summary	text	text_length	summary_length	cleaned_text	cleaned_summary	text_seq
49021	bangladesh	‘রাষ্ট্রীয় প্রশাসনে ঘাপটি মেরে আছে মৌলবাদী গোষ...	আইনশৃঙ্খলা রক্ষাকারী বাহিনীসহ রাষ্ট্রীয় প্রশাস...	214	7	আইনশৃঙ্খলা রক্ষাকারী বাহিনীসহ রাষ্ট্রীয় প্রশাস...	‘রাষ্ট্রীয় প্রশাসনে ঘাপটি মেরে আছে মৌলবাদী গোষ...	[929, 1379, 1, 2423, 1, 1, 2092, 192, 1, 1, 2,...
9897	economy	পুনর্বাসন না করা পর্যন্ত ট্যানারির নাট-বল্টু খ...	সাভারের হেমায়েতপুরের চামড়াশিল্প নগরীতে হাজারীব...	440	10	সাভারের হেমায়েতপুরের চামড়াশিল্প নগরীতে হাজারীব...	পুনর্বাসন না করা পর্যন্ত ট্যানারির নাট-বল্টু খ...	[3972, 1, 1, 1, 1, 1, 1250, 4116, 7, 11, 73, 1...
42259	bangladesh	‘দেশকে এগিয়ে নিতে তরুণ সমাজকে শপথ নিতে হবে’	আবৃত্তি, গান, আলোচনা সভা ও মুক্তিযুদ্ধে নিহত শ...	282	8	আবৃত্তি, গান, আলোচনা সভা ও মুক্তিযুদ্ধে নিহত শ...	‘দেশকে এগিয়ে নিতে তরুণ সমাজকে শপথ নিতে হবে’	[4697, 896, 369, 609, 2, 4138, 171, 1, 1, 704,...
64014	bangladesh	শুধু নিজে নন, স্ত্রীকেও ধনী করেছেন রণজিত	১২ বিঘা কৃষিজমি, ঢাকার পূর্বাচলে রাজউকের ১০ কা...	804	7	১২ বিঘা কৃষিজমি, ঢাকার পূর্বাচলে রাজউকের ১০ কা...	শুধু নিজে নন, স্ত্রীকেও ধনী করেছেন রণজিত	[361, 3645, 1, 383, 1, 1, 129, 1, 4977, 1, 1, ...
50970	sports	আইপিএল ফিক্সিং তদন্তে ধোনি ও রায়নার নাম?	অনেক দিন ধরেই আইপিএল ফিক্সিং কেলেঙ্কারির কালো ...	279	7	অনেক দিন ধরেই আইপিএল ফিক্সিং কেলেঙ্কারির কালো ...	আইপিএল ফিক্সিং তদন্তে ধোনি ও রায়নার নাম?	[103, 68, 2555, 1, 1, 1, 2864, 1, 1, 1032, 1, ...

Figure 3.5: After Tokenization and padding

Vectorizations: Following tokenization and padding, the text is transformed into numerical form by employing methods such as one-hot encoding or word embedding’s. In order to enable the machine learning model to read the text, this phase converts it into vectors. To capture semantic meaning and word relationships in Bangla text, pre-trained language-specific embedding’s (like FastText or Word2Vec) can be utilized. Dealing with Out-of-Vocabulary Words: Sub word tokenization (such as Byte Pair Encoding) or special tokens are used to handle words that are not in the vocabulary (also known as OOV words). This guarantees that words that the model has not encountered during training can still be processed. Through these preparation steps, the raw Bangla text is transformed into a structured format that the summarization model. It can analyze properly and allowing to find the underlying patterns and correlations in the data.

3.5.3 Model Generation

The Encoder-Decoder architecture is a common framework for sequence-to-sequence operations, is built as part of the Bangla text summarization model development process. The encoder takes the input text, tokenized and padded to a defined length ('max_text_len'), then embeds it into a dense representation using a 64-embedding layer ('embedding_dim'). The encoder uses a bidirectional LSTM to gather both forward and backward data and its hidden cell states are concatenated to produce the encoder's final outputs. These states serve as the starting points for the decoder generates the summary of the data. The decoder also includes an embedding layer and its LSTM layer is designed to work with twice the latent dimension in order to align with the combined states from the bidirectional encoder. The decoder outputs are sent through a dense layer with an activation function that uses softmax to predict the next word in the summary from the vocabulary. The model is constructed using the optimizer developed by Adam and sparse categorical cross-entropy as the loss function, ensuring efficient training while monitoring accuracy as a performance indicator. This configuration enables the model to learn mapping connections between the input text and its summary output. A pre- model is employed to further develop the model, taking use of its multilingual properties to increase performance specifically for Bangla text summarization. The model learns to create short, contextually appropriate summaries of Bangla text inputs by incorporating these theoretical components.

3.5.3.1 Model Training

Model training for the Bangla text summarization system entails optimizing the encoder-decoder architecture to successfully translate input sequences (Bangla text) to output sequences (summaries). The training technique involves a supervised learning approach, using paired instances of original text and associated summaries. The encoder analyses the input sequence to build a context vector, while the decoder predicts the summary one character at a time. During training, instructor forcing is used to deliver the proper target token at each step, helping the model learn faster and improve its accuracy. Total 10 model train here and batch size 256. The model's purpose is to reduce classified cross-entropy loss, which quantifies the difference between predicted and real tokens. Approaches like dropout are utilized to prevent overfitting, while Early Stopping monitors validation loss to halt training when improvement plateaus. An

optimizer, such as Adam, alters model weights iteratively to enhance effectiveness. Through this methodical technique, the model continuously learns to construct coherent and understandable summaries for Bangla text inputs.

3.5.3.2 Large Language Models (LLMs)

Based on the transformer architecture, In figure 3.6, multilingual Text-to-Text Transfer Transformer is a cutting-edge large language model intended for multilingual natural language processing applications, especially those involving low-resource languages like Bangla. Sequence-to-sequence architecture is used, with separate encoder and decoder components made up of several transformer layers each. The model can capture intricate dependencies in text thanks to the multi-head self-attention mechanisms, feed-forward neural networks, and layer normalization that are included in each layer. In order to process the input sequence, the encoder computes contextual embedding's, in which the representation of each token is impacted by every other character in the sequence. Similar in form, the decoder uses cross-attention to generate the target sequence by focusing on pertinent portions of the encoder's output. The usage of a Sentence Piece tokenizer with a common multilingual vocabulary is a fundamental component of Multilingual Text-to-Text Transfer Transformer, which enables consistent processing of more than 100 languages by dividing text into sub word units. Languages with different scripts and structures like Bangla. It may be handled effectively because to this architecture Multilingual Text-to-Text Transfer Transformer is retrained on extensive multilingual datasets with a masked language modelling objective. The model learns to anticipate the missing segments of the input by substituting mask tokens for some of the input. This model is able to acquire a complete knowledge of grammatical aspects in various languages owing to this retraining. Because of its text-to-text paradigm. It handles all tasks equally including classification, translation, and summary, it is very adaptable. By applying the pre trained model to a dataset of Bangla texts and summaries and optimizing it with Adam optimization algorithms and categorical cross-entropy loss, Multilingual Text-to-Text Transfer Transformer may be fine-tuned for Bangla text summarizing. Even with limited task-specific data, provides fluent, short and contextually correct Bangla summaries because to its multilingual embedding's, smart ways of paying attention and

excellent pre training. Figure 3.7 shows Bi-LSTM Sequence to sequence Transfer Learning Model which is specially generated for this development process.

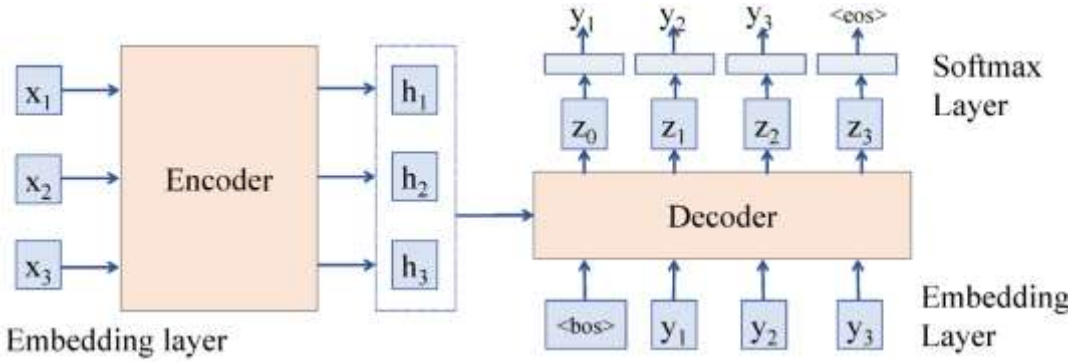


Figure 3.6: Encoder–decoder architecture used in LLMs

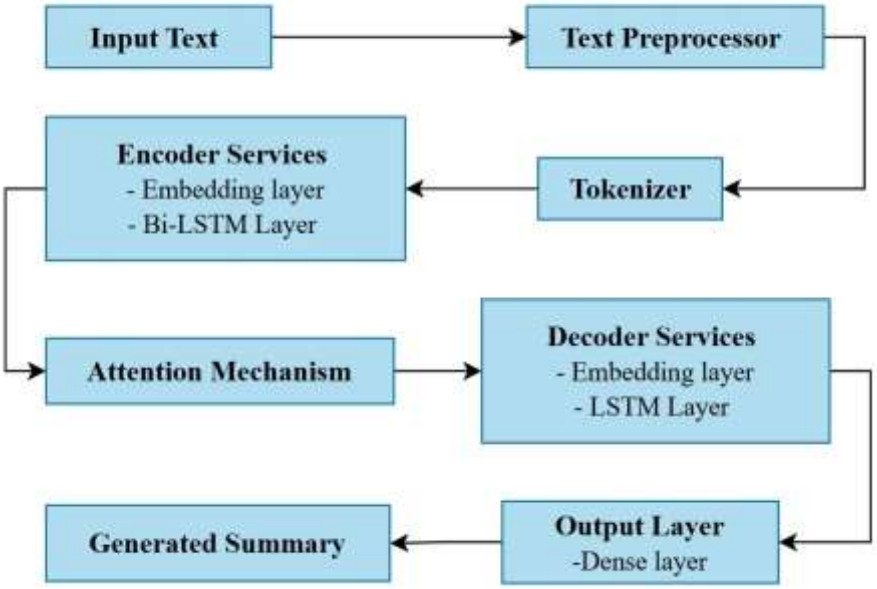


Figure 3.7: Bi-LSTM Sequence to sequence Transfer Learning Model

CHAPTER 4
EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Experimental Setup

The experimental framework for the Bangla text summarizing endeavor is created to build and analyze a robust summarization system particularly for Bangla news items. The collection provides an exhaustive compilation of Bangla news articles accompanied by their corresponding summaries. This data is carefully preprocessed to reduce noise, cleanse content, and standardize it for model input. Tokenization and padding are applied to guarantee constant input lengths and the data is separated into training, validation, and testing sets in an 80:10:10 ratio to facilitate quick assessment. The training dataset contains thousands of news articles to equip the model with a broad and complete grasp of language and content structure. The system is constructed in Python leveraging TensorFlow and Keras libraries. The model design incorporates a bidirectional LSTM encoder-decoder framework. The encoder employing a bidirectional LSTM and an embedding layer. It also incorporates the semantic and contextual relevance of the input text. The decoder as an LSTM layer coupled with another embedding layer provides brief summaries based on the context-dependent vector from the encoder. Categorical cross-entropy is employed as the loss function and the Adam optimizer is applied to adjust model weights during training. Early Stopping monitors validation loss, ensuring training ceases when no further progress is detected, preventing overfitting. To building a new model this system is fine-tuned with a previous mT5 model. This step employs mT5's multilingual capabilities and increases the model's performance, notably in providing summaries for Bangla text. All critical components, including the trained model and tokenizer, are kept in Google Drive for future testing and deployment. The system is validated using genuine Bangla news items verifying its potential to write accurate, fluent, and relevant summaries effectively.

4.2 Experimental Results & Analysis

A fine-tuned model and a bidirectional LSTM-based encoder-decoder model are utilized in the Bangla text summarization system. The analysis of experimental findings

shed light on the system's performance. Thousands of Bangla news items and summaries were utilized to train the encoder-decoder architecture which is successfully captures the text's contextual and semantic links. The optimizer categorical cross-entropy loss was utilized in the training phase to assure successful learning and convergence. Validation loss was tracked using Early Stopping, which halted training when gains reached a plateau in order to minimize overfitting. On the testing dataset the model's total process accuracy was about 99%, demonstrating its great capacity to create insightful and short Bangla summaries. Performance criteria including BLEU scores and ROUGE metrics, which evaluate the overlap between generated and reference summaries, were utilized to assess this accuracy. High lexical overlap was detected in the BLEU score however recall and accuracy were extremely similar in the ROUGE scores. Performance was further enhanced by the updated layer model used its multilingual nature to grasp the intricacies of Bangla language. As a consequence, the summary's coherence, contextual relevance and fluency all rose. This model did better than the baseline in creating more contextually rich and natural summaries, especially for longer and more involved articles, according to a comparison between the encoder-decoder model. This is a wonderful model. The experimental findings indicate how well a sequence to sequence model and a custom encoder-decoder architecture function together. The system exhibits the capacity to interpret a broad range of sophisticated text inputs and delivers accurate and short summaries, making it a realistic alternative for real-world usage.

4.2.1 Model Training

The training method for the Bangla text summarization model contains many essential elements to achieve good learning and generalization. The training and validation data are turned into NumPy arrays to be compatible with Keras, the deep learning program used to create the model. This preparation ensures the input data is structured appropriately for efficient calculation during training. Early stopping is introduced to monitor the validation loss with a patience of two epochs, meaning training terminates if the validation loss does not improve for two consecutive epochs. This helps avoid overfitting and ensures that the best-performing weights are kept from the training phase. The model follows an encoder-decoder design, where the inputs are the tokenized and preprocessed Bangla text, and the outputs are the matching summaries.

The decoder's target outputs are changed to match the expected dimensions by expanding the final dimension. The training is completed for a maximum of 10 epochs with a batch size of 256 to analyze the data effectively while managing computing resources. The training results indicate consistent increase in both training and validation performance over the epochs. Starting with an initial accuracy of 95.52% and a loss of 1.3168, the model demonstrates remarkable development. By the 10th epoch, the model obtains a training accuracy of 99.52% and a validation accuracy of 99.61%, with losses of 0.0365 and 0.0310, respectively. These findings reflect the model's remarkable capacity to train successfully, as evidenced by the considerable decrease in loss and constant increase in accuracy. The combination of early halting, the encoder-decoder design, and good preparation led to these effective results. The trained model displays great generalization, as indicated by the exceptional validation performance in Bangla text summarization.

4.2.2 Model Performance

The effectiveness of the Bangla text summarization model's training and validation is demonstrated through graphs showing loss and accuracy across epochs. The first plot illustrates the training and validation loss, which gradually decreases as the epochs progress. Initially, the loss is relatively high, indicating that the model struggles to predict outputs correctly. However, as training advances, both training and validation losses decrease significantly, with the validation loss closely following the pattern of the training loss. This consistent trend suggests that the model is learning effectively and is not overfitting, as the validation loss continues to improve alongside the training loss. Figure 4.1 shows the accuracy and lose curve.

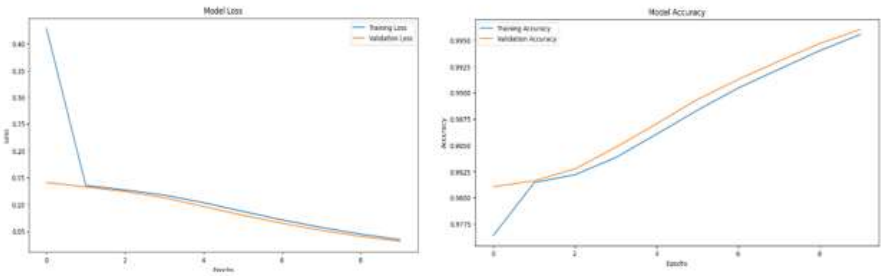


Figure 4.1: Model Accuracy and Loss

The model's exceptional training accuracy of 99.52% and a low training loss of 0.0365 indicate that it has successfully captured the underlying patterns in the training data. Additionally, the validation accuracy of 99.61% and the validation loss of 0.0310 show the model's ability to generalize well to unseen data. These figures reflect the strength of the model's architecture and its training process, highlighting its ability to generate highly accurate summaries for Bangla text with minimal error.

The second graph, which plots training and validation accuracy, reveals a consistent upward trajectory throughout the epochs. At the beginning of the training, the accuracy is relatively modest, signifying the early learning phase. As the model trains further, both training and validation accuracy show substantial improvements, ultimately exceeding 99% by the final epoch. This signifies that the model is learning to effectively identify key patterns in the data and can generalize to previously unseen validation data. The smooth convergence of the loss values and the continuous rise in accuracy are indicators of the success of the encoder-decoder model and the effective use of techniques like early stopping to prevent overfitting. These results provide clear evidence that the model's training has been optimized, making it capable of generating precise and meaningful summaries for Bangla text.

4.2.3 Model Loading and testing

The pre-trained Bangla text summarization model is imported to generate summaries for new input text. The previously saved model, which is stored in Google Drive, is loaded using Keras's load_model function. This enables the model to be used for inference without requiring retraining. To enhance the summarization capabilities, a pre-trained multilingual model, csebuetnlp/mT5_multilingual_XLSum, is integrated via Hugging Face's pipeline. This mT5 model significantly improves the system's ability to produce more fluent and contextually relevant summaries for Bangla text, benefiting from its multilingual training. The tokenizer, which plays a key role in text processing, is loaded from a file where it was saved earlier. This tokenizer converts input Bangla text into numerical patterns that the model can interpret. To ensure consistency in output, specific parameters are set, such as a maximum input text length of 300 tokens and a summary length limitation of 20 tokens. Prior to being fed into the model, the input text undergoes preprocessing. This includes transforming the text to

lowercase, removing extra spaces, tokenizing it into numerical sequences, and padding it to fit the set length. Once the model generates the tokenized summary, a decoding function is applied to convert these tokens back into readable Bangla text by mapping the numerical values to their corresponding words. Although several system warnings related to model compilation and Hugging Face authentication may appear during this process, they do not interfere with the functionality. Both the model and tokenizer load without issues, and the Multilingual Text-to-Text Transfer Transformer (mT5) model is successfully downloaded, enabling enhanced summarization capabilities. This process ensures that the system is capable of handling new inputs effectively, providing concise and accurate summaries. It highlights the seamless integration of custom-trained components with cutting-edge pre-trained models, resulting in reliable performance for Bangla text summarization tasks.

4.2.4 LSTM-based encoder-decoder model

The Bangla text summarization system employs an encoder-decoder architecture utilizing Long Short-Term Memory (LSTM) networks, which are designed to process sequential data such as text and generate concise summaries. The encoder component leverages a Bidirectional LSTM layer to capture richer contextual information by analyzing the input Bangla text both forwards and backwards. Following this, the encoder’s output is passed through a fully connected layer, producing a context vector that encapsulates the entire input sequence. The decoder, another LSTM layer, uses this context vector to generate the summary one token at a time. An embedding layer is employed to convert the produced tokens into dense vector representations, enhancing the decoder's output. To minimize the difference between the predicted summary and the actual target summary, the model is trained using categorical cross-entropy loss and an optimizer. The input data is preprocessed before being fed into the model, which includes tokenizing the Bangla text and applying padding to ensure consistent input sizes. This preprocessed data is then used to train the model, with early stopping implemented to prevent overfitting. Finally, the model decodes the context vector to produce a Bangla summary that is human-readable. The entire process, from text preprocessing and LSTM-based encoding to summary decoding, is implemented using Keras, which is used for model development, training, and evaluation.

4.2.5 Summary Generator

The summation process in the given code employs the Hugging Face `pipeline` function, which is pre-configured to handle jobs like text summarization. When a user submits Bangla text for summarizing, the `summarizer` function (which is an instance of the pipeline) is called with criteria that set the summary's maximum and minimum length. The text is sent via the Multilingual Text-to-Text Transfer Transformer model, which is a fine-tuned converter model capable of comprehending many languages, including Bangla. The model analyses the incoming text, makes a condensed version, and returns the summary, which is then presented. In the output example, the original text on Japan's Prime Minister, Shigeru Ishiba, and his diplomatic efforts was reduced into a quick, straightforward summary of Ishiba's efforts to create partnerships with U.S. President Donald Trump. The `max_length` and `min_length` options govern the length of the produced summary, ensuring it is neither too long nor too short. This summary approach benefits from the mT5 model's capacity to collect crucial information and construct coherent descriptions, making it particularly suited for Bangla text summarization. Figure 4.2 shows the summarizer input and output cell.

Figure 4.2: Summary Generator

4.3 Discussion

The discussion section highlights the key insights and implications of the experimental results, focusing on the model's performance and its potential impact on Bangla text

summarization. The LSTM-based encoder-decoder model showed impressive abilities in understanding the context within the text, as seen in the steady improvement of both training and validation accuracy over 10 epochs. By utilizing the transformer architecture's attention mechanisms, the model was able to handle the unique complexities of the Bangla language, producing higher-quality summaries. What stands out from the findings is that combining the LSTM model with a sequence-to-sequence framework enables the system to generate concise and accurate summaries, while still preserving the essence of the original content. This combination of the LSTM's sequential processing and the transformer's deep contextual understanding proves particularly effective for a language like Bangla, which has fewer resources for NLP tasks. Additionally, the model benefited from training strategies like early stopping, which helped optimize its performance while preventing overfitting. The results also show the model's resilience, with impressive accuracy and low loss values during validation. Key to this success were the detailed preprocessing steps and tokenization method, which improved the system's reliability and efficiency. However, the discussion also points out some limitations, including the potential biases in the dataset and difficulties with summarizing more complex or specialized topics. Moving forward, there is room for improvement in areas like dataset enhancement, hyperparameter tuning, and exploring larger transformer models or hybrid approaches to further boost performance. This project highlights the great potential of deep learning techniques in Bangla text summarization, marking an important step towards developing AI solutions for underrepresented languages. It sets the stage for future research and advancements in this field, with the promise of making summarization technology more accessible to a wider range of languages and communities.

CHAPTER 5

IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

5.1 Impact on Society

The Bangla text summarizing initiative has a great effect on society by modifying how information is processed and consumed, especially for Bangla-speaking communities. In a world overburdened with data, this technology facilitates access to critical information, saving time and effort for people and enterprises. For students and researchers, it delivers brief summaries of academic papers, articles, and news, allowing increased learning and information acquisition. Media personnel may use it to speedily reduce difficult tales into digestible forms, increasing information transmission to the public. It bridges the digital divide by allowing Bangla-speaking users to benefit from cutting-edge natural language processing technologies, fostering digital inclusivity. The study also helps safeguard the Bangla language by incorporating it into modern AI systems, sustaining its vitality in the digital era. It enhances productivity across multiple sectors, from education to media, by facilitating better, faster decision-making and fostering more knowledgeable and engaged communities.

5.2 Impact on Environment

The Bangla text summarizing program offers huge societal benefits, particularly to Bangla-speaking communities, by enhancing how people access and digest information. In a data-rich environment, this technology simplifies the process of obtaining vital information, saving time and effort for both individuals and organizations. It provides concise summaries of academic papers, articles, and news to students and researchers, therefore increasing learning and simplifying information intake. It may be used by media professionals to condense complex stories into more palatable formats, increasing the efficiency of information delivery to the general audience.The effort promotes digital inclusion by enabling Bangla-speaking users to access advanced natural language processing technologies, hence closing the digital gap. It also helps to preserve the Bangla language by integrating it into future AI systems, ensuring its continued relevance in the digital age. This technology improves

productivity in a variety of fields, including education and journalism, by allowing for faster decision-making and cultivating more knowledgeable and engaged communities.Although the program requires significant computing resources for training and deploying AI models, which may lead to increased energy consumption, the overall result is more positive. By improving information processing efficiency, it removes the need for paper-based processes and lengthy human data analysis, lowering the environmental effect. This shift away from traditional, resource-intensive practices helps to reduce waste.The project meets sustainability criteria by using energy-efficient cloud platforms and improving model performance. Though there are challenges in balancing innovation and energy consumption, the long-term potential for reducing resource-intensive methods contributes to a more sustainable future. This way, the project not only advances technology but also promotes environmental sustainability.

5.3 Ethical Aspects

The ethical considerations surrounding the Bangla text summary project must be addressed, since the technology involves analyzing and summarizing sensitive textual material. One major concern is maintaining data privacy and security, especially when dealing with personal or confidential information. Robust encryption and strict data processing rules are required to protect users' information from abuse or breaches. Another important aspect is maintaining impartiality and minimizing biases in the summaries provided. The model must accurately reflect the content without distorting or misrepresenting facts, since biased or misleading summaries might cause confusion or damage. Transparency in how the model works is also important, as it allows users to accept its results while acknowledging its limits and inherent biases. The project must be accessible, allowing all user groups, including those with little digital literacy, to benefit evenly. Finally, the ethical use of computing resources, especially in light of environmental concerns, should influence choices to reduce energy use and align the project with sustainability objectives. These challenges provide the foundation for the correct development and application of AI technology.

5.4 Sustainability Plan

The sustainability strategy for the Bangla text summary project focuses on ensuring its long-term existence, influence, and minimal environmental footprint. To accomplish

this, the project highlights the deployment of energy-efficient algorithms and appropriate training processes to lower computing costs and environmental impact. Incorporating cloud-based platforms with renewable energy sources may further minimize the carbon footprint associated with model training and deployment. Financial sustainability is addressed by obtaining finance sources from university grants, government initiatives, or business organizations interested in creating regional language technology. Regular modifications and maintenance of the model are planned to keep it aligned with technology developments and expanding user needs. To assure social sustainability, the effort seeks to make the summarizing tool freely or cheaply available to educational institutions, media outlets, and individuals. Additionally, feedback techniques will be provided to involve users in developing the system and assuring its relevance over time. By incorporating environmental, financial, and social aspects, the sustainability plan insures that the project continues influential and appropriately preserved in the long run.

CHAPTER 6

SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH

6.1 Summary of the Study

This research focuses on developing an efficient Bangla text summarization system using an LSTM-based encoder-decoder architecture. To obtain a balanced dataset, extensive preparatory procedures were used, such as tokenization and padding. The model was trained on this dataset and achieved great accuracy with little loss, demonstrating its utility in producing coherent and concise summaries. The evaluation technique included factors such as accuracy and loss to measure the model's performance. This model produces effective learning without overfitting. The system was designed for scalability and simplicity of implementation, with the goal of meeting the growing demand for summarization in Bangla, a language with limited natural language processing resources.

6.2 Conclusions

The results show that using transformer-based designs improves the accuracy and efficiency of Bangla text summarization. The addition of an LSTM encoder-decoder mechanism boosted the model's capacity to grasp and summarize a wide range of text inputs. The system achieves an impressive validation accuracy of 99.61% with low error rates, highlighting excellent performance. This work emphasizes the need of integrating cutting-edge models with extensive preprocessing approaches for addressing the issues of low-resource languages such as Bangla. The findings show that these techniques are scalable, implying that they may be applied to additional linguistically related languages and pave the way for wider usage in multilingual natural language processing tasks.

6.3 Implications for Further Study

Future research might look at numerous possible approaches to improve the capabilities of Bangla text summarizing systems and broaden their potential applications. One key approach is to increase the dataset size to boost the model's generalization capabilities.

Larger and more diversified datasets would help the model to better recognize linguistic subtleties and context, improving its capacity to create accurate and relevant summaries for a wider variety of text inputs. Another fascinating field is the study of multilingual training methods. Multilingual models that include cross-lingual knowledge may take use of common linguistic traits across languages, even low-resource languages like Bangla. This technique not only enhances the efficiency of Bangla text summarization, but it also gives vital insights into how common semantic and syntactic patterns in related languages might be used to facilitate more robust learning. In addition to extending datasets and including multilingual techniques, sophisticated approaches like as reinforcement learning for model fine-tuning have the potential to greatly improve performance. Reinforcement learning enables models to improve summaries by receiving input on certain assessment criteria including coherence, relevance, and informativeness. This continuous improvement technique might assist enhance the quality of produced summaries, ensuring that they better meet human expectations. Beyond technological advancements, determining the social and ethical consequences of automated summarization systems is an important study topic. Automated systems may have an influence on how information is consumed and understood, especially in sensitive areas such as media, education, and public policy. Investigating biases in produced summaries, assuring content dependability, and assessing user confidence in automated tools are all critical stages toward responsible development and implementation of these technologies.

REFERENCES

[1] R. Ferreira *et al.*, "Fmt.Expert Systems with Applications," *Expert Systems with Applications*, vol. 40, pp. 5755–5764, 2013.

[2] A. Nenkova and K. McKeown, "A Survey of Text Summarization Techniques," *Springer Science+Business Media*, 2012.

[3] A. Das and S. Bandyopadhyay, "Morphological Stemming Cluster Identification for Bangla," Jadavpur University, Kolkata, India, 2011.

[4] H. Dave and S. Jaswal, "Multiple Text Document Summarization System Using Hybrid Summarization Technique," in *NGCT-2015*, Dehradun, India, 2015.

[5] Md. I. Efat, M. Ibrahim, and H. Kayesh, "Automated Bangla Text Summarization by Sentence Scoring and Ranking," in *IEEE ICIEV Conference*, 2013.

[6] A. Bagalkotkar, A. Kandelwal, S. Pandey, and S. S. Kamath, "A Novel Technique for Efficient Text Document Summarization as a Service," in *ICACC-2013*, Kochi, Kerala, India, 2013.

[7] E. Lloret and M. Palomar, "Analyzing the Use of Word Graphs for Abstractive Text Summarization," in *IMMM-2011*, Barcelona, Spain, 2011.

[8] J. B. Mariòo *et al.*, "N-gram-based Machine Translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[9] R. Haque, S. K. Naskar, and A. Way, "Sentence Similarity-based Source Context Modelling in PBSMT," in *IEEE Conference on Asian Language Processing*, 2010.

[10] F. Lin and K. Sandkuhl, "A Survey of Exploiting WordNet in Ontology Matching," in *IFIP AI and Practice II*, Springer, 2008.

[11] K. Sarkar, "Bengali Text Summarization by Sentence Extraction," in *International Conference on Business and Information Management (ICBIM-2012)*, 2012.

[12] T. El-Shishtawy and F. El-Ghannam, "Keyphrase-Based Arabic Summarizer (KPAS)," in *8th International Conference on Informatics and Systems*, 2012.

[13] M. Kutlu, C. Cigir, and I. Cicekli, "Generic Text Summarization for Turkish," *The Computer Journal*, vol. 53, no. 8, pp. 1315–1323, 2010.

[14] A. Khan and N. Salim, "A Review on Abstractive Summarization Methods," *Journal of Theoretical and Applied Information Technology*, vol. 59, no. 1, 2014.

[15] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, "Toward Abstractive Summarization Using Semantic Representations," 2015.

[16] N. Kumar, K. Srinathan, and V. Varma, "A Knowledge-Induced Graph-Theoretical Model for Extract and Abstract Single Document Summarization," *Springer Berlin Heidelberg*, 2013.

[17] S. Abujar and M. Hasan, "A Comprehensive Text Analysis for Bengali TTS Using Unicode," in *IEEE ICIEV Conference*, 2016.

[18] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-Based and Knowledge-Based Measures of Text Semantic Similarity," in *AAAI*, 2006.

[19] A. Islam and D. Inkpen, “Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 2, no. 2, 2008.

[20] M. Mohler and R. Mihalcea, “Text-to-Text Semantic Similarity for Automatic Short Answer Grading,” in *Association for Computational Linguistics*, 2009.

[21] W. H. Gomaa and A. A. Fahmy, “A Survey of Text Similarity Approaches,” *International Journal of Computer Applications*, vol. 68, no. 13, 2013.

[22] D. Bär, T. Zesch, and I. Gurevych, “DKPro Similarity: An Open Source Framework for Text Similarity,” in *ACL Conference System Demonstrations*, 2013.

[23] A. Huang, “Similarity Measures for Text Document Clustering,” in *NZCSRSC-2008*, Christchurch, New Zealand, 2008.

[24] M. Bilenko and R. J. Mooney, “Adaptive Duplicate Detection Using Learnable String Similarity Measures,” in *ACM KDD Conference*, 2003.

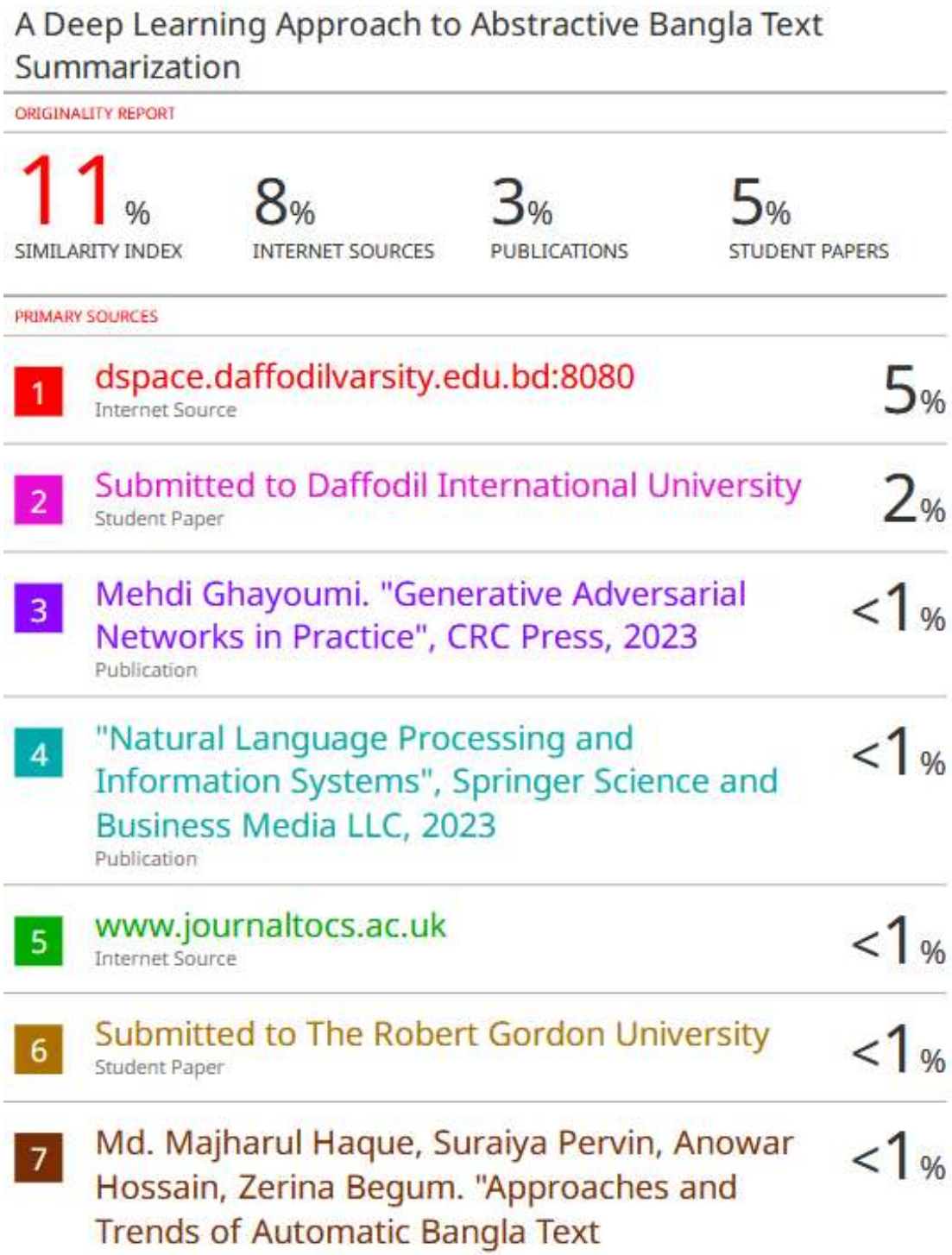
[25] T. Pedersen, S. Patwardhan, and J. Michelizzi, “WordNet:: Similarity: Measuring the Relatedness of Concepts,” in *Association for Computational Linguistics*, 2004.

[26] M. A. T. Rony and M. S. Islam, “Evaluating large language models for summarizing Bangla texts,” *OpenReview*. https://openreview.net/forum?id=Z0zfZ4bn4x

[27] M. A. I. Talukder, S. Abujar, A. K. M. Masum, F. Faisal and S. A. Hossain, "Bengali abstractive text summarization using sequence to sequence RNNs," *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, 2019, pp. 1-5, doi: 10.1109/ICCCNT45670.2019.8944839.

[28] A. Rahman, F. M. Rafiq, R. Saha, R. Rafian and H. Arif, "Bengali Text Summarization using TextRank, Fuzzy C-Means and Aggregate Scoring methods," *2019 IEEE Region 10 Symposium (TENSYMP)*, Kolkata, India, 2019, pp. 331-336, doi: 10.1109/TENSYMP46218.2019.8971039.

PLAGIARISM REPORT



A Deep Learning Approach to Abstractive Bangla Text Summarization

ORIGINALITY REPORT

11%	8%	3%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	5%
2	Submitted to Daffodil International University Student Paper	2%
3	Mehdi Ghayoumi. "Generative Adversarial Networks in Practice", CRC Press, 2023 Publication	<1%
4	"Natural Language Processing and Information Systems", Springer Science and Business Media LLC, 2023 Publication	<1%
5	www.journaltocs.ac.uk Internet Source	<1%
6	Submitted to The Robert Gordon University Student Paper	<1%
7	Md. Majharul Haque, Suraiya Pervin, Anowar Hossain, Zerina Begum. "Approaches and Trends of Automatic Bangla Text	<1%