AUTOMATIC HINDI NEWS HEADLINE GENERATION USING TRANSFORMER BASED LANGUAGE MODELS

A thesis submitted in partial fulfilment of the requirements
for the award of the degree
of

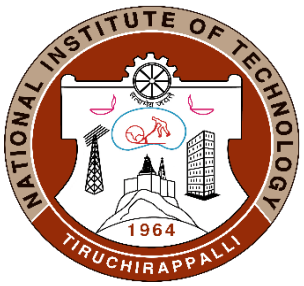
M. Tech

in

Data Analytics

By

GAURAV THAKUR (205222008)



DEPARTMENT OF COMPUTER APPLICATIONS
NATIONAL INSTITUTE OF TECHNOLOGY
TIRUCHIRAPPALLI - 620015

JUNE 2024

BONA FIDE CERTIFICATE

This is to certify that the project work titled **"AUTOMATIC HINDI NEWS HEADLINE GENERATION USING TRANSFORMER BASED LANGUAGE MODELS"** is a bona fide record of the work done by

GAURAV THAKUR (205222008)

in partial fulfilment of the requirements for the award of the degree of **Master of Technology** in **Data Analytics** of the **National Institute of Technology, Tiruchirappalli,** during the year 2022-2024.

Dr. Vishnu Priya R	**Dr. Michael Arock**
Project Guide	Head of the Department

Project Viva-voce held on _____

Internal Examiner	**External Examiner**

ABSTRACT

Natural language processing (NLP) is seeing rapid progress in the domain of automatic text summarization, particularly in light of the exponential expansion of digital data. One major problem in natural language understanding and information retrieval is distilling information without sacrificing its meaning. Extractive summarization techniques preserve accuracy and grammar by rephrasing and copying sections from the original text. Conversely, by adding new terms to the original text, abstractive summarization algorithms create new phrases. Although most of the prior research has been on extractive techniques, abstractive frameworks provide sophisticated capabilities like generalization and paraphrase.

News headline generation, a subset of abstractive text summarization, plays a crucial role in converting lengthy texts into concise summaries. With just a few lines of text, it provides a summary, saving time for readers. However, text summarization, especially for languages other than English, remains challenging due to limited resources and data, particularly for languages like Hindi.

This study aims to develop an automated text summarization system using state-of-the-art machine learning models. The Transformer-based models, including a variant of BART and Multilingual Text-To-Text Transfer Transformer (mT5), were fine-tuned on the Hindi Text Short Summarization Corpus. BART outperforms mT5 by 15.77% in rogue-1 scores. After evaluation the Rogue-1 metrics which includes precision, f1-score and recall for the fine-tuned models IndicBART and mt5 are (0.52,0.54,0.57) and (0.37,0.39,0.40) respectively. Also the BLEU score are 0.13 and 0.08 respectively. This indicates how automatic summarization systems could be improved, and how Hindi-speaking people could benefit from better material retrieval and understanding.

ACKNOWLEDGEMENT

I would like to extend my gratitude to **Dr. G. Aghila, Director**, National Institute of Technology, Tiruchirappalli for having provided all the facilities to carry out this project work.

I would like to thank **Dr. Michael Arock, Head of the Department**, Department of Computer Applications, National Institute of Technology, Tiruchirappalli for allowing me to avail the facilities of the department during project work.

I express my sincere thanks and gratitude to my guide **Dr. Vishnu Priya R, Assistant Professor**, Department of Computer Applications, National Institute of Technology, Tiruchirappalli who directed me from time to time throughout the project with valuable guidance, constant encouragement, and suggestions.

My hearty thanks to all the members of the Project Coordination Committee, Department of Computer Applications, National Institute of Technology, Tiruchirappalli, for their valuable suggestions during the project reviews.

I am also thankful to the faculty, staff members and research scholars of the Department of Computer Applications, National Institute of Technology, Tiruchirappalli, for their support and suggestions throughout the project.

Gaurav Thakur

(205222008)

TABLE OF CONTENTS

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BART	Bidirectional AutoRegressive Transformer
T5	Text To Text Transfer Transformer
ROGUE	Recall Oriented Understudy for Gisting Evaluation
BLEU	Bilingual Evaluation Understudy
NLP	Natural Language Processing
CPU	Central Processing Unit
GPU	Graphics Processing Unit
ATS	Automatic Text Summarization

CHAPTER 1
INTRODUCTION

The internet is overflowing with information in the current digital era, which causes information overload for many consumers. To address this issue, automatic summarization systems have been developed to provide users with concise summaries of various sources. These systems can summarize single articles, clusters of news articles, broadcast news shows, or email threads, distilling the most important information. Applications for summarizing many kinds of information, including news, medical information, scientific publications, and discussions, have proliferated in recent years. While these systems are not flawless, they have proven to be beneficial to users and have enhanced other automatic applications and interfaces. However, the availability of processed text data is currently limited to certain languages, primarily English, which restricts the reach of natural language processing (NLP) applications. To make NLP applications accessible to a wider audience, it is essential to develop NLP systems for a diverse range of languages. This thesis seeks to emphasize the need for the development of text summarization resources and systems specifically tailored for Indian languages.

1.1 Motivation

In today's digital age, the abundance of online information often overwhelms users, making it challenging to access and process relevant content. There is a rising need for tools that can summarize enormous amounts of text succinctly in order to address this problem. Automatic text summarization systems offer a solution by condensing lengthy documents into shorter, more digestible summaries. These systems are useful for various types of content, including news articles, scientific papers, and emails, allowing users to quickly grasp the main points.

While such systems are effective for English and some other languages, there's a noticeable gap in resources and systems for Indian languages like Hindi. This lack of support limits the accessibility of NLP applications to a significant portion of the population. This project is motivated by the need to fill this gap and make NLP technologies more inclusive for Indian language speakers. By developing text summarization resources and systems tailored for Indian languages, we aim to empower users to engage with digital content more effectively in their native languages.

Additionally, this project contributes to efforts to preserve linguistic diversity and cultural heritage in the digital realm. Investing in NLP systems for Indian languages enriches linguistic resources and facilitates access to information for language communities.

Overall, this project aims to address the challenges of information overload and language accessibility, creating a more inclusive digital environment where users can access and understand content in their preferred languages.

1.2 Typology Of Summaries

1.2.1 Source Oriented

Single vs. Multi-document: While multi-document summarizing entails combining data from several thematically related publications into a summary, single-document summarization is usually used to give a succinct synopsis of a single text. Multi document summarization is valuable when trying to gain a comprehensive understanding of a topic or event by merging information from various sources. The main difference between single and multi document summarization lies in the size and scope of the input or source documents.

Domain-specific vs. General: Domain-specific summarization generates summaries from one or more documents focused on a particular domain or field. These models are tailored to capture the unique terminologies, jargon, and writing styles specific to that domain. Conversely, general domain summarization can summarize input text from any domain, without being restricted to a specific field or subject matter.

1.2.2 Target Oriented

Extractive vs. Abstractive Summaries: Extractive summaries are generated by selecting important snippets, sentences, or passages directly from the source document. These selections are taken verbatim from the original text. Conversely, abstractive summaries aim to convey relevant information by rephrasing and restructuring phrases or clauses, using the summary author's own words.

Guided vs. Controllable Summarization: Guided summarization involves utilizing various guidance signals to minimize the deviation of the summary from the source document. It incorporates additional information to steer the summarization process, allowing for controllability through user-specified inputs. Controllable summarization, on the other hand, gives users the ability to modify important features of the generated summary, according to their favorite information source, including style, length, and the

emphasis placed on particular items.

1.2.3 Purpose Oriented

Generic vs. Query-focused summaries: A generic summary presents the author's perspective of the input text by giving equal weight to all aspects of the source material. In contrast, a query-focused summary prioritizes specific themes or aspects based on the user's interest in learning about a particular topic. This type of summary may exclude certain themes or aspects to directly address the user's query. To generate a useful summary in this context, the summarizer model must consider both the query and the document.

Indicative vs. Informative: An indicative summary offers a high-level overview of the input text without covering all its contents. It provides a general indication of the text's topic without specific details or supporting evidence. On the other hand, an informative summary enables the reader to understand the main points of the input article, although not necessarily all its content. Informative summaries include specific details, data, and examples from the original content and are typically written in a formal and objective tone.

Role-based or Multi-perspective: A multi-perspective summary provides a summary of a given text focusing on specific themes or aspects. For example, there may be several accurate summaries, each offering a distinct viewpoint, in a scientific paper. One summary might focus on the "abstract and introduction" sections, while another could concentrate on the "results and conclusions."

1.3 Problem Statement

This project's main goal is to create a reliable text summarizing system designed especially for Hindi. This system aims to comprehend users input in Hindi language, effectively retrieve pertinent information, and deliver precise and grammatically accurate summaries (news headlines) in Hindi. The emphasis lies on creating a seamless user experience where users can interact with the system and receive accurate and coherent summaries, thereby enhancing accessibility to information and improving the overall user satisfaction in Hindi-speaking communities.

To achieve this goal, the system will incorporate advanced natural language processing techniques, including text processing, information retrieval in Hindi. The system's design and implementation will prioritize accuracy, efficiency, and linguistic correctness, ensuring that users can trust the system to provide reliable Hindi news headlines.

1.4 Objectives

- Develop a robust summarization system tailored for the Hindi language.
- For text summarization, fine-tune and assess how well the pre-trained Transformer-based language models perform on the Hindi dataset.
- Evaluate the performance of the summarization system using metrics such as Rogue score and Bleu score to assess its effectiveness in handling Hindi inputs and providing satisfactory summaries.
- Contributing to the development of NLP tools for Hindi: The initiative may aid in the creation of NLP instruments that are applicable to a range of Hindi-language applications.

1.5 Report Outline

This chapter summarized the introduction to the problem of information overload in the digital age, overview of automatic summarization systems and their importance in addressing this issue, discussion on the proliferation of summarization applications and their limitations, especially for Indian languages like Hindi. Explanation of the motivation behind the project, emphasizing the need for text summarization systems tailored for Indian languages. Objective behind the project. Importance of linguistic diversity and cultural heritage preservation in the digital realm. The rest of the report is organized as follows:

- Chapter 2 presents the already existing solutions and their drawbacks.
- Chapter 3 presents the proposed solution.
- Chapter 4 presents the implementation details along with the results obtained.
- Chapter 5 concludes the work done and future work.

CHAPTER 2
LITERATURE SURVEY

Text summarization in low resource languages has long been a persistent challenge in the field of NLP. There are distinct methods for abstractive and extractive summarization, but they share common features such as salience and coverage. Various approaches have been employed for summarizing Indian languages, ranging from statistical and linguistic-based techniques to pure machine learning and deep learning methods.

2.1 Summarization Approaches

2.1.1 Extractive Summarization:

The main challenge in extractive summarization is to effectively select and organize important sentences. Initially, heuristic-based methods [17], frequency-based techniques, and clustering approaches [32] were used for summarization. However, the k-means extraction method suffered from out-of-order extraction, although the summaries generated through the frequency-based approach were more fluent. To address the issues with sentence ranking, TextRank [6] and PageRank [17] algorithms were applied. Several works have attempted multi-document summarization, as described in the works of [2] and [4]. In order to investigate semantic relations on Hindi documents, the extractive summarizer of [1] developed the ATS (Automatic Text Summarization) model based on document vector approach work. It is worth noting that most summarization systems for Indian languages are designed for extractive summarization, as it is relatively easier to implement. Examples of such extractive summarizers can be found in [30, 8, 27, 15, 25, 26, 7, 9].

2.1.2 Abstractive Summarization:

Abstractive summarization methods are known to generate more fluent and coherent summaries compared to extractive methods. In earlier work on abstractive summarizers for Indian languages, information extraction techniques [31] and automatic keyword extraction methods [3] were used. These summarizers utilize parts-of-speech tagging to create headlines and comprise various components, such as word cues, summary generation, sentence extraction, keyword extraction, and sentence selection modules. These components analyze the textual data to identify the main features of the summary. Neural attention models [34], Seq2Seq RNNs [35], and Pointer-Generator networks [12] are a few examples of recent

abstractive approaches to summarizing that concentrate on creating summaries that capture the sense of the input text without necessarily selecting sentences directly from the text. Abstractive techniques are more common and provide high-quality summaries with the advent of big neural language models for generation tasks. [36] provides an explanation of how to fine-tune the BART and T5 transformers for abstractive summarization. Although there have been numerous advancements in model architectures and summarizing methods, the availability of large-scale datasets like CNN/DailyMail [35, 28], Gigaword [12, 23], XSum [14], etc. has been largely responsible for the progress made in English text summarization. Research on the application of abstractive text summarizing techniques is scarce. Despite utilizing a smaller dataset for training, Bhargava et al. [18] investigated the use of generative adversarial networks for multilingual text summarization and achieved performance levels equivalent to previous techniques. An attention-based stacking long short-term memory model (LSTM) was proposed by Singh et al. [37] and produced effective short summaries. In 2022, Shah et al. [38] presented a summarization technique employing deep learning, specifically using LSTM neural networks and word embeddings. Their study leveraged word embeddings to convert text into vector representations, with LSTM cells forming the core of the encoder and decoder components in Seq2Seq networks.

2.1.3 Guided and Controlled Summarization:

Abstractive summaries, while they may sound fluent, often suffer from issues due to their unconstrained nature. These issues include unfaithful summaries that may contain factual errors or hallucinated content. Moreover, it is challenging to train models to focus on specific aspects and control the summaries. To address these problems, a recent study by Dou et al. (2021) [18], proposed guided summarization as a solution. This approach constrains the output generation of models to minimize deviation from the original source and allows for controllability through user-specified inputs. Previous research has also explored ways to guide neural abstractive summarization models. For instance, Kikuchi et al. (2016) [33] focused on specifying the length of summaries, while Li et al. (2018) [20] provided models with keywords to avoid omitting crucial information. Cao et al. (2018) [11] proposed models that retrieve and reference relevant summaries from the training set. In their recent work, Dou et al. (2021) [18] introduced a framework to investigate four types of guidance signals, including highlighted sentences, keywords, salient relation triples (subject, relation, object), and retrieved summaries.

2.1.4 Multi-perspective Summarization:

It is important to remember that there is a lot of variance in the amount of material presented [24], the degree of depth, and the way the content is arranged in the summaries of scientific documents. A recent effort by Fabbri [19] builds the multi-perspective answer summary corpus using topic threads from the Yahoo forum. Meng et al. [29] give FactSum, which has four summaries for every publication that address various topics. Summaries can be provided upon request from users. Neural models ScisummNet [21] and SciTLDR for extreme summarization [10] were trained on several academic document summarizing datasets, such as PubMed and arXiv [13]. A multi-perspective summarizing dataset for scientific documents was released by the organizers of the multi-perspective summarization shared task, in contrast to these datasets. In summarization tasks, several generation models have demonstrated excellent performance, such as BART, T5, ProphetNet, and PEGASUS. Models like Big Bird [22] and Longformer [5] were released to handle long documents.

CHAPTER 3
PROPOSED WORK

3.1 Architecture:

Transformers are different from other architectures in that they can parallelize model training because their primary foundation is attention mechanisms. Furthermore, unlike the seq2seq paradigm, the encoder-decoder does not require an RNN, LSTM, or CNN. Two transformer-based pre trained models (mT5 and IndicBART) are used in this work.

3.1.1 Multilingual Based BART Model (IndicBART)

BART, a novel transformer model that was unveiled in 2019, combines an autoregressive GPT decoder with a two-way BERT encoder to maximize pretraining learning. Facebook AI created the BART (Bidirectional and Auto-Regressive Transformer) language model, of which IndicBART is a variation. It is intended especially for activities involving the production of text in Indian languages, such as Hindi. Large-scale datasets in several Indian languages have been used to pre-train IndicBART.

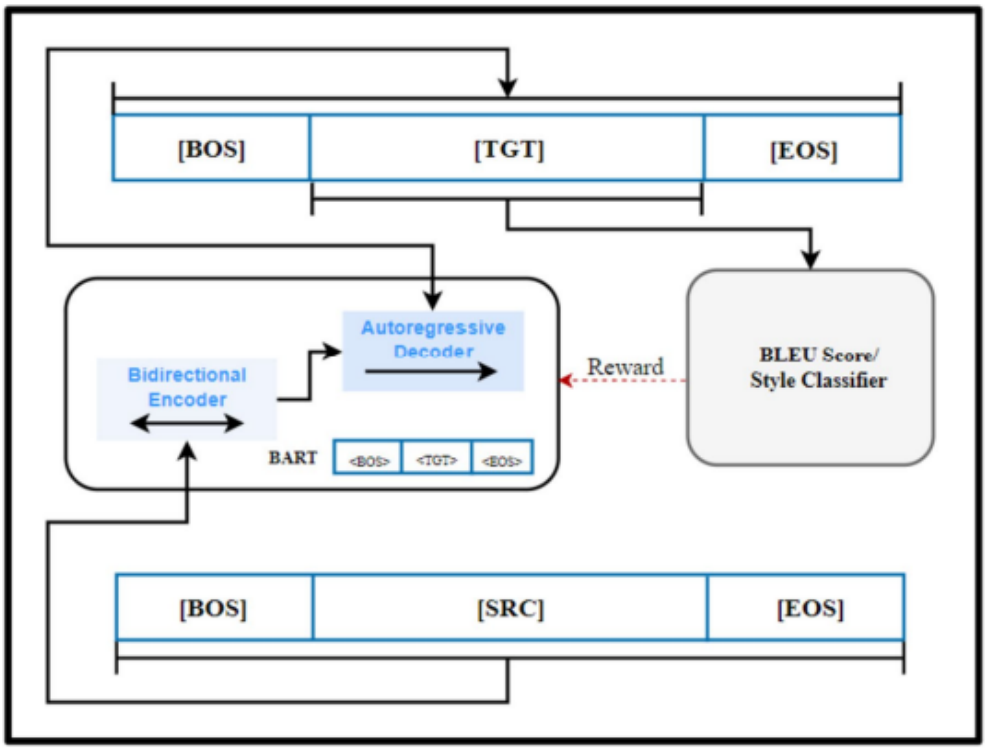


Figure 1: BART Architecture

This model leverages the power of the transformer architecture to capture contextual

information effectively, allowing it to generate high-quality summaries, translations, and other text outputs in Hindi and other Indian languages. IndicBART addresses the need for NLP tools tailored to Indian languages, helping to bridge the gap in language technology and make NLP applications more accessible to Hindi-speaking communities.

When presented with a piece of text in Hindi, IndicBART first encodes the input text using its transformer-based architecture. This encoding process captures the semantic and syntactic information of the input text. Once the input text is encoded, IndicBART generates a summary by decoding the encoded representation into a sequence of words. Using an autoregressive generation method, it makes predictions about the following word in the summary by looking at the context of the words that have already been formed.

3.1.2 Multilingual Based T5 Model (mT5)

It is a multilingual variant of the "text to text transfer transformer" or T5 developed by Google AI. It operates on the text-to-text principle, meaning it frames all NLP tasks as a text-to-text mapping problem. mT5 is designed to be language agnostic, meaning it can handle languages with varying grammar, syntax, and vocabulary without requiring language-specific modifications. This makes it highly adaptable to diverse linguistic environments.

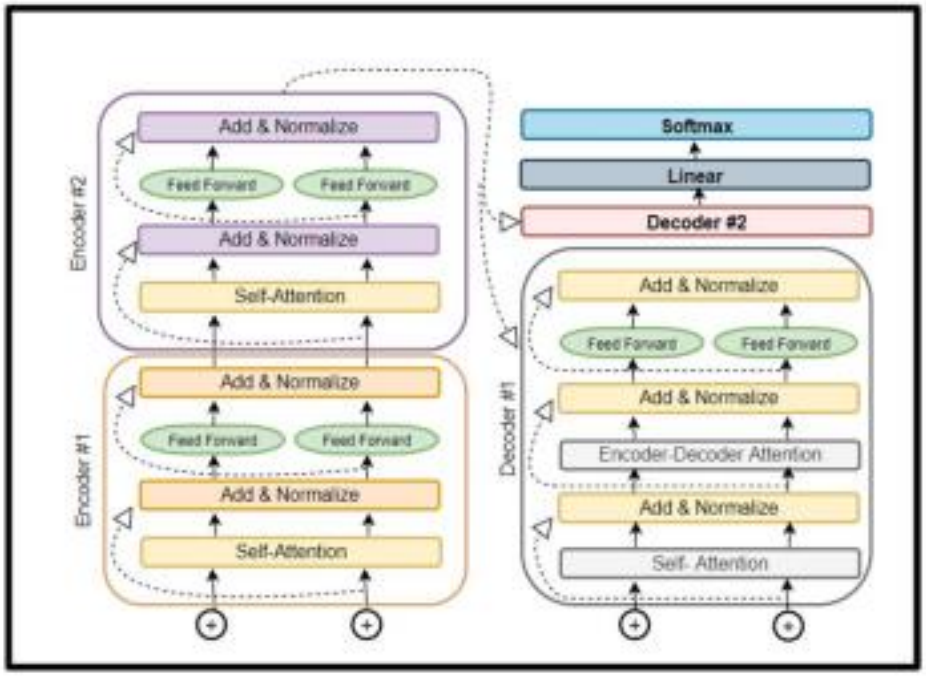


Figure 2: T5 Architecture

mT5 is pre-trained on a large corpus of text data from multiple languages, including but not limited to English, Hindi, Spanish, French, German, and many others. During pre-training, the model learns to understand and generate text in different languages, capturing language-specific patterns and semantics. Similar to its forerunner T5, mT5 adheres to the text-to-text framework, which formulates all NLP tasks—summarization included—as text-to-text tasks. This suggests that the input and output of the model are represented by text strings. With the usage of an encoder-decoder architecture, mT5 generates the matching summary by first converting the input text into a fixed-size vector representation, which is then used by the decoder.

Model	Number of encoder layers	Number of decoder layers	Number of attention heads	Trainable Parameters (millions)
Bart	12	12	16	406
T5	6	6	8	80

Table 1: Model Architecture

Generalized Equations for Transformer-based Models (mT5 and IndicBART).

The transformer model for generating Hindi news headlines follows a structured approach involving both encoding and decoding stages, where each stage is composed of multiple layers that perform specific functions.

Encoder Layer

The encoder processes the input text through several layers. Each layer involves two key operations:

1. Self-attention mechanism: As a result, the model can concentrate on various sections of the input text.

$$\text{Attention} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Here:

$-$ (Q) (Query) is derived from the input.

$-$ (K) (Key) is also derived from the input.

$-$ (V) (Value) represents the input values.

$-(d_k)$ is the dimension of the key vectors.

2. Feed-Forward Network (FFN): This transforms the data through a series of linear operations and a non-linear activation function (ReLU).

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2$$

Here:

$-$ (x) is the input.

$-(W_1)$ and (W_2) are weight matrices.

$-(b_1)$ and (b_2) are bias terms.

The output of each encoder layer is then normalized and combined:

$$\text{Layer Output} = \text{LayerNorm}(x + \text{Attention} + \text{FFN})$$

Decoder Layer

The decoder processes the data in multiple layers, each performing three main operations:

1. Masked Self-Attention: This mechanism ensures the model does not look ahead in the sequence it is generating.

$$\text{MaskedAttention} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + \text{Mask}\right)V$$

2. Cross-Attention: As a result, the decoder can concentrate on incorporating contextual information into the encoder's output.

$$\text{CrossAttention} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

3. Feed-Forward Network (FFN): Similar to the encoder's FFN.

$$\text{FFN}(y) = \text{ReLU}(yW_1 + b_1)W_2 + b_2$$

The output of each decoder layer is also normalized and combined:

$$\text{Layer Output} = \text{LayerNorm}(y + \text{MaskedAttention} + \text{CrossAttention} + \text{FFN})$$

Final Output

Following processing through every layer, the decoder applies a linear transformation and then uses a softmax function to predict the next word in the sequence, producing the final output:

$$\text{Output} = \text{softmax}(y_L W_o + b_o)$$

Summary of Notations

$-InputText((x))$: The Hindi text input to the encoder.

$-EncoderOutput((x_L))$: The final encoded representation of the input Hindi text.

$-GeneratedSequence((y))$: The sequence of words generated by the decoder so far.

$-DecoderOutput((y_L))$: The final output of the decoder layers.

$-Weights((W_o))$ and $Bias((b_o))$: Parameters of the final linear transformation layer.

Simplified Equations

$$\begin{aligned} &\text{Encoder:} && x_L = \text{EncoderLayers}(x) \\ &\text{Decoder:} && y_L = \text{DecoderLayers}(y, x_L) \\ &\text{Output:} && \text{Headline} = \text{softmax}(y_L W_o + b_o) \end{aligned}$$

Here, The probability distribution for the next word in the headline is generated using (x) the Hindi input text, (x_L) the encoded representation from the encoder, (y) the partially generated headline, (y_L) the final output from the softmax, and the decoder.

3.2 Dataset:

index	headline	article
0	EXCLUSIVE: दिल्ली में डीज़ल टैक्सियों पर बैन से मुश्किल में पड़ा चुनाव आयोग	दिल्ली में सुप्रीम कोर्ट के डीज़ल टैक्सियों को बंद करने के फैसले के बाद हजारों टैक्सी ड्राइवरों की रोजी रोटी पर तो असर पड़ा ही है, लेकिन अब दिल्ली पर एक और नई मुसीबत आ गई है. चुनाव आयोग राजधानी के 13 वाई में उपचुनाव करवा रहा है, लेकिन चुनावों से दो हफ्ते पहले चुनाव आयोग में कामकाज ठप्प हो गया है. कमीशन ने किराए पर ली थी डीजल गाड़ियां दरअसल कमीशन ने लगभग सौ गाड़ियां चुनाव के कामकाज को करने के लिए किराए पर लीं, जिनमें सभी डीज़ल से चलने वाली टैक्सी थीं. इन्हीं टैक्सियों से चुनाव अधिकारी से लेकर चुनावों का जिम्मा संभालने वाले बाकी कर्मचारी भी एक जगह से दूसरी जगह आते जाते थे. अचानक चुनावों से ठीक पहले आई इस परेशानी ने दिल्ली चुनाव आयोग का कामकाज ही ठप्प कर दिया है. रियायत के लिए की जा सकती है मांग दिल्ली के राज्य चुनाव अधिकारी राकेश मेहता ने इस मुश्किल का रास्ता निकालने के लिए मंगलवार को दिल्ली के पुलिस कमिश्नर और ट्रांसपोर्ट कमिश्नर की बैठक बुलाई है. इस बैठक में राज्य चुनाव आयुक्त 15 मई को होने वाले चुनावों को लेकर गाड़ियों की उपलब्धता को लेकर पुलिस और सरकार से समाधान निकालने के लिए भी कहेंगे. चुनाव आयोग इन दोनों एजेंसियों को कह सकता है कि चूंकि चुनाव में अब दो हफ्ते का भी वक्त नहीं बचा है, ऐसे में इन गाड़ियों को बैन से रियायत दी जाए.
1	जॉर्डन: राष्ट्रपति मुखर्जी ने 86 करोड़ डॉलर के संयंत्र का उद्घाटन किया	जॉर्डन के ऐतिहासिक दौरे पर पहुंचे राष्ट्रपति प्रणब मुखर्जी ने 86 करोड़ डॉलर की लागत से निर्मित भारत-जॉर्डन उर्वरक संयंत्र का जॉर्डन के शाह अब्दुल्ला (द्वितीय) इब्न अल हुसैन के साथ उद्घाटन किया. यह संयंत्र एक साल से कम समय में बनकर तैयार हुआ है. राष्ट्रपति मुखर्जी के यहां ऐयर इंडिया की उड़ान से दोपहर पहुंचने के कुछ देर बाद ही शाह के महल से इस संयंत्र का रिमोट के जरिए उद्घाटन किया गया. अधिकारियों ने बताया कि भारतीय उर्वरक कंपनी इफको और जॉर्डन के फास्फेट्स माइन कंपनी ने इस संयंत्र के लिए 2008 में एक संयुक्त उद्यम कंपनी जॉर्डन इंडिया फर्टिलाइजर कंपनी बनाया. संयुक्त उद्यम में इफको की हिस्सेदारी 52 प्रतिशत है. इस संयंत्र से प्रति वर्ष 450 करोड़ टन सल्फ्यूरिक एसिड और 150 करोड़ टन फास्फोरिक एसिड के उत्पादन का अनुमान है. राष्ट्रपति का इससे पहले यहां पारंपरिक स्वागत किया गया और राष्ट्रपति भवन के सामने उन्हें 21 तोपों की सलामी दी गई. इसके बाद वह शाह अब्दुल्ला (द्वितीय) इब्न अल हुसैन के साथ वार्ता में व्यस्त हो गए. वार्ता के बाद दोनों नेताओं ने इंडो-जर्मन उर्वरक संयंत्र का संयुक्त रूप से उद्घाटन किया. इस संयंत्र से कच्चे माल का उत्पादन किया जाएगा. इसमें फास्फोरिक एसिड और सल्फ्यूरिक एसिड प्रमुख हैं. यहां पहुंचने से पहले राष्ट्रपति ने कहा, 'दोनों देशों के क्षेत्रीय एवं अंतर्राष्ट्रीय मुद्दे मिलते-जुलते हैं और दोनों सीरिया के साथ ही मध्य पूर्व में शांति प्रक्रिया का समर्थन करते हैं.' उन्होंने कहा कि दोनों देश उग्रवाद और आतंकवाद के सभी रूपों की निंदा करते हैं और धार्मिक सौहार्द में भरोसा करते हैं. राष्ट्रपति के इस दौरे के दौरान व्यापार एवं निवेश पर भी जोर है. उन्होंने कहा कि दोनों देश द्विपक्षीय व्यापार को पांच अरब डॉलर करना चाहते हैं. अभी दोनों देशों के बीच व्यापार दो अरब डॉलर है. प्रणब मुखर्जी ने जिस संयंत्र का उद्घाटन किया, उससे भारत 30 करोड़ टन फास्फोरिक एसिड का आयात करेगी. भारत बड़ी मात्रा में पोटाश एवं फास्फेट जॉर्डन से हासिल करता है. भारत और जॉर्डन ने 1947 में सामंजस्य के लिए द्विपक्षीय समझौते पर हस्ताक्षर किया था, हालांकि इसे औपचारिक रूप 1950 में दिया गया जब पूर्ण कूटनीकि संबंध दोनों देश के बीच बने. शाह अब्दुल्ला और बेगम रानिया ने अक्टूबर 2012 में भारत का दौरा किया था. राष्ट्रपति के इस दौरे से पूर्व करीब 30 साल पहले तत्कालीन प्रधानमंत्री राजीव गांधी ने इस देश का दौरा किया था. इनपुट-IANS
2	UN में पाकिस्तान की राजदूत मलीहा लोधी ने कराई फजीहत, मांगनी पड़ी माफी	पाकिस्तानी नेताओं को विवादित और हास्यास्पद बयान आए दिन सुर्खियां बंटोरते रहते हैं और कई बार तो पाकिस्तान के मंत्री तक भड़काऊ बयान देने से बाज नहीं आते. इस कड़ी में संयुक्त राष्ट्र में पाकिस्तान की प्रतिनिधि का नाम भी जुड़ गया है जिन्होंने ब्रिटेन के प्रधानमंत्री बोरिस जॉनसन को ब्रिटिश विदेश मंत्री बता दिया. मलीहा लोधी ने एक फोटो ट्वीट की जिसमें पाकिस्तान प्रधानमंत्री इमरान खान और ब्रिटिश प्रधानमंत्री बोरिस जॉनसन की मुलाकात हो रही है. इस फोटो का कैप्शन लिखते वक्त लोधी से चूक हो गई और वह जॉनसन को ब्रिटेन का विदेश मंत्री लिख बैठीं. इसके कुछ देर बाद ही उन्हें अपनी गलती का अहसास हुआ और उन्होंने अपना ट्वीट डिलीट कर दिया. मलीहा लोधी का ट्वीट पाकिस्तान की राजदूत ने फिर से ट्वीट करते हुए अपने पुराने ट्वीट पर माफी मांगी और कहा कि टाइप करने में गलती हो गई. प्रधानमंत्री इमरान ने ब्रिटिश पीएम से मुलाकात की है. Sorry typo in previous tweet. Prime Minister Imran Khan Met British PM this morning. pic.twitter.com/Ufp9vz5Ent — Maleeha Lodhi (@LodhiMaleeha) September 23, 2019 बता दें कि पाकिस्तान के प्रधानमंत्री इमरान खान संयुक्त राष्ट्र महासभा के 74वें सत्र में हिस्सा लेने अमेरिका गए हुए हैं. यहां इमरान खान ने कई नेताओं से मुलाकात की और कश्मीर को लेकर बातचीत है. पाकिस्तान के प्रधानमंत्री सोमवार रात को राष्ट्रपति डोनाल्ड ट्रंप से मुलाकात करेंगे जहां वह ट्रंप से कश्मीर के मसले पर मध्यस्थता करने की गुहार फिर से लगाने वाले हैं.
	38 देशों में पीएम [illegible]	पीएम नरेंद्र मोदी बायोपिक में विवेक ओबेरॉय ने प्रधानमंत्री नरेंद्र मोदी का किरदार निभाया है. जानकारी के अनुसार फिल्म भारत के अलावा कई देशों में रिलीज हो सकती है. प्रोड्यूसर और डिस्ट्रीब्यूटर आनंद पंडित ने बताया कि पीएम नरेंद्र मोदी बायोपिक की टीम फिल्म को 38 देशों में रिलीज करने का प्लान बना रही है. इसमें यूएस, यूके, कनाडा, ऑस्ट्रेलिया और यूएई जैसे देश शामिल हैं. पिंकविला के अनुसार प्रोड्यूसर आनंद ने बताया कि पीएम नरेंद्र मोदी की लाइफ को लेकर देश ही नहीं विदेशों के ऑडियंस में ही गजब की रुचि है. इसलिए इसे इंडिया के अलावा 38 देशों में रिलीज करनी योजना है. उन्होंने कहा, भारत में यह फिल्म 1700 स्क्रीन पर रिलीज होगी. इसके अलावा इसे विदेशों में लगभग 600 स्क्रीन पर रिलीज करने की तैयारी की जा रही है. मोदी बायोपिक को बैन करने की मांग पर आनंद पंडित ने कहा, "फिल्म को लेकर कोई शक नहीं होना चाहिए. सभी जानते हैं कि ये फिल्म क्या है. यह बायोपिक एक सिनेमेटिक प्रोडक्ट है. जो लोग इस फिल्म को लेकर सवाल उठा रहे हैं और इस पर बैन लगाने की मांग कर रहे हैं. दरअसल, वे लोग अभिव्यक्ति की स्वतंत्रता को दबाना चाहते हैं. क्या यह पाखंड नहीं है कि जो लोग दूसरी फिल्मों को बैन करने की आतोचना करते हैं वे ही आज मोदी बायोपिक को बैन करने की मांग कर रहे

✓ Connected to Python 3 Google Compute Engine backend (TPU)

Figure 3: Snapshot of dataset.

The Hindi Text Short Summarization Corpus comprises approximately 330,000 articles paired with their corresponding headlines, sourced from Hindi news websites. The dataset is split into two subsets: a test set with roughly 66,000 articles and a training set with about 266,000 articles.

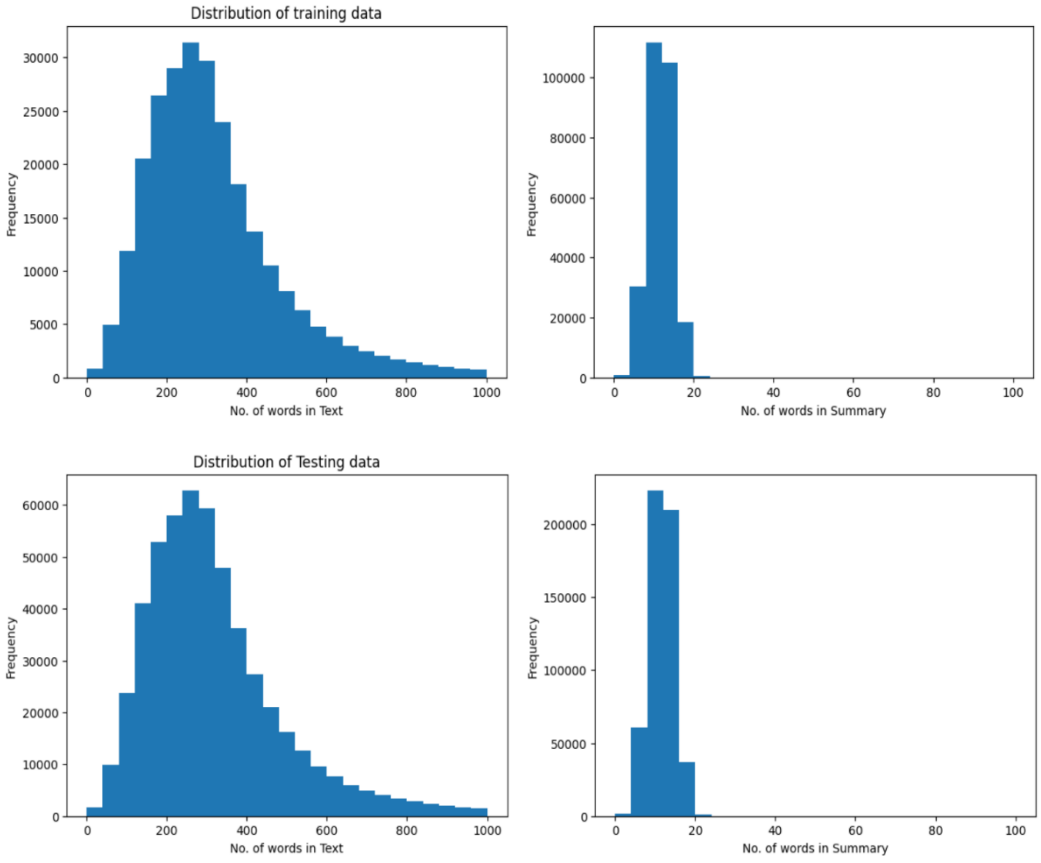


Figure 4:Distribution of train and test data based on word counts

3.3 Preprocessing:

3.3.1 Dataset Cleaning

Dataset cleaning is a vital component of the data preprocessing pipeline, especially for natural language processing tasks like text summarization.

- Emoticons, symbols, and pictographs, as well as Chinese characters, transport symbols, HTML tags, and URLs, are removed from the text.
- Punctuation marks and special characters are stripped from the sentences to ensure uniformity and clarity.
- Whitespace cleaning is performed to eliminate blank spaces, which is a standard step in preparing the dataset for further processing.

3.3.2 Tokenization

Tokenization is the process of dividing text into smaller pieces called tokens. Depending on the chosen tokenization strategy, these tokens can be words, sub words, or characters. By converting text into tokens, the model can analyze and understand the text's semantic and syntactic properties, enabling more accurate and meaningful summaries. Better handling of linguistic subtleties is made easier by this procedure, which also guarantees that the model can produce accurate and pertinent summaries based on the input data.

3.3.3 Padding

Padding is the fundamental technique in NLP used to ensure uniformity in sequence lengths within a dataset. This is particularly crucial for neural networks, which generally require inputs of a fixed size. By adding padding tokens to shorter sequences, all sequences can be made to match the length of the longest sequence in the dataset.

3.3.4 Word Embeddings

Word Embedding is an NLP technique that converts word or phrases from a given vocabulary into continuous vector space representations. These vectors encapsulate the semantic meanings and interrelationships of words, ensuring that terms having comparable meanings have comparable vector representations.

CHAPTER 4
IMPLEMENTATION AND RESULTS

4.1 Implementation details

System used for the experiments has the following configuration:

- Processor — Intel i5 7th Generation 1.6 GHz – 3.4 GHz
- RAM — 8 GB
- System Type — 64-bit
- Operating System — Windows 11
- GPU — Nvidia 940Mx – 2GB

4.2 Software Environment

The following software tools and libraries were used:

- IDE — Google Collaboratory, Kaggle Notebook
- Programming Language — Python
- Libraries — NumPy, PyTorch, Pandas.
- GPU — NVIDIA Tesla T4 GPUs
- RAM — 12.7 GB

4.3 Procedure details:

4.3.1 Fine tuning pre-trained model:

- **BART based pre-trained model (IndicBART):**

During the fine tuning process of the IndicBART model, several key parameters and strategies were employed to optimize its performance for the abstractive summarization task on Hindi news. The learning rate was set to 1e-3, ensuring a gradual and stable adjustment of model weights during training. With a training epoch of 5, the model underwent multiple iterations to learn task-specific patterns from the labeled data. The training process resulted in a training loss of 0.33, indicating the convergence and generalization capabilities of the model.

To manage the training data effectively, a batch size of 4 was used in combination with the AdamW optimizer. This setup facilitated efficient updates to the model's parameters and helped in mitigating overfitting. The data was organized into dataloaders, with a random

split strategy allocating 30% for testing, 70% of the data for training. This split ensured a representative distribution of data across training and test sets allowing for robust model evaluation.

After fine-tuning, the model achieved an rogue-1 score of of 0.54 and bleu score of 0.13, indicating its ability to generate accurate Hindi news headlines and perform well on the evaluation metrics. These scores demonstrate the effectiveness of the fine tuning process in enhancing the model's performance for the summarization task it was trained on.

Byte-pair Encoding Tokenizer:

IndicBART utilizes a Byte pair encoding (BPE) tokenizer, which is a subword tokenization algorithm that breaks down text into subword units based on the frequency of character sequences. This tokenizer is specifically designed for Indian languages like Hindi, enabling effective text processing and generation. Trained on extensive datasets across multiple Indian languages, it accurately captures the unique features and structures of these languages. The BPE tokenizer in IndicBART is adept at handling the complexities of Indian languages, including compound words and inflectional suffixes. By tokenizing text into subword units, the model can process and generate Hindi text while preserving its nuances. Overall, the BPE tokenizer in IndicBART facilitates accurate and efficient text summarization and other NLP tasks for Indian languages.

- **Multilingual text-to-text transfer transformer (mT5):**

During the fine tuning of the mT5 model, a systematic approach was adopted to improve its functionality for the abstractive summarization task on Hindi corpus. The procedure starts by loading a cleaned dataset and splitting it into training, test, and validation sets. The model is then initialized with the mT5 checkpoint, and a custom dataset class is created to preprocess the data and tokenize it using the sentencepiece tokenizer. The data is loaded into dataloaders for efficient batch processing during training. The model's configuration includes parameters like learning rate of 1e-3, 5 training epochs, and a batch size of 4. During training, the AdamW optimizer is employed to update the model's parameters, whereas the cross-entropy loss function is used to process the dataset and calculate the discrepancy between the expected and actual summaries. The activation function used in the model is the GPT2-style activation function, which is a variant of the ReLU activation function. Additionally, the dataset is split into training, test and validation sets with a split of 70%, 5%, and 25%, respectively, to ensure representative distribution and robust

evaluation. Finally, the trained model achieves satisfactory performance metrics, indicating its effectiveness in generating accurate summaries in Hindi.

The model's performance metrics, including an rogue-1 score of 0.39 and an bleu score of 0.09, demonstrated its accuracy and efficacy in generating summaries and excelling on evaluation metrics.

Sentencepiece Tokenizer:

The Sentencepiece algorithm-based tokenizer is used by mT5, or Multilingual Text-to-Text Transfer Transformer. Sentencepiece is an unsupervised text tokenizer and detokenizer designed mainly for text generation systems based on neural networks that need to partition words into subwords. It breaks down text into subword units, allowing it to effectively handle various languages and character sets.

mT5's tokenizer is specifically designed to efficiently handle multilingual text, enabling the model to process text in multiple languages seamlessly. By tokenizing input text into subword units, mT5 can understand and generate text across different languages using a single unified vocabulary. This capability allows mT5 to perform tasks like translation, summarization, and question answering in multiple languages.

The SentencePiece tokenizer offers high customizability, allowing users to specify parameters such as vocabulary size and tokenization algorithm. This flexibility enables mT5 to adapt to different languages and text types, making it a versatile tool for multilingual NLP tasks.

4.3.2 Optimizer :

- **AdamW optimizer:**

AdamW is an extension of the Adam optimizer, which stands for Adaptive Moment Estimation with Weight Decay. It combines the principles of momentum and adaptive learning rates to efficiently update model parameters during training. Here's how AdamW works:

1. Adaptive learning rates: AdamW adjust the learning rate for each parameters individually, allowing for quick convergence while preventing large oscillations in the loss function. It computes adaptive learning rates based on the 1st and 2nd moments of the gradients.

2. Momentum: AdamW incorporates momentum to accelerate optimization by maintaining an exponentially decaying average of past gradients' 1st and 2nd moments (Mean and uncentered variance).

3. Weight decay: Weight decay, a regularization method that penalizes high weights by including a term in the loss function proportionate to the squared size of the weights, is denoted by the "W" in AdamW. Smaller weights are encouraged, therefore overfitting is discouraged.

4. Bias Correction: AdamW applies bias correction to address potential biases in estimates of the first and second moments of gradients, particularly during early training stages. This correction ensures unbiased estimates and enhances optimization stability.

All things considered, AdamW is a good optimizer for deep neural network training since it strikes a compromise between regularization, momentum, and adaptability. It is extensively utilized in many different machine learning applications, including as reinforcement learning, computer vision, and natural language processing.

4.4 Evaluation Metrics:

4.4.1 Rogue Score:

The ROUGE Score, which contrasts the produced text with the reference summary, serves as the foundation for evaluating the Summarization model. An n-gram is a sequence of n words, and ROUGE-N counts the co-occurrences of n-grams between the generated and reference summaries. ROUGE-2, for example, assesses the correspondences between subsequent word pairs in both summaries. Equation (1) is used to compute the ROUGE-N metric.

$$ROUGE\text{-}N = \frac{\sum_{S_e} \sum_{S_h} \sum_{g_n \in S} C_m(g_n)}{\sum_{S_e} \sum_{S_h} \sum_{g_n \in S} C(g_n)} \qquad (1)$$

— S_h denotes a collection of candidate summaries.
— S_e refers to a specific candidate summary.
— g_n signifies an N-gram.

— $C(g_n)$ represents the total occurrences of g_n in both the candidate and the reference summaries.

The precision metric is the ratio of the total words in the candidate summary to the number of matched words in the reference and candidate summaries. Recall evaluates how well the N-grams from the reference summary are captured in the candidate summary. Recall gauges the extent to which the summary is thorough, whereas precision shows the relevance. Both measures are necessary to assess how well the anticipated summary performs. Consequently, the harmonic mean of precision (P) and recall (R) is used to get the F-score:

$$F1\text{-score} = 2 \times \frac{\mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}} \qquad (2)$$

4.4.2 Bleu Score:

The BLEU score evaluates the degree of similarity between machine-generated translations and reference translations through the examination of n-grams, which are word sequences consisting of one or more words, such as unigrams, bigrams, trigrams, and so on. By contrasting these n-grams in the machine translation with those in the reference translations, it assesses the accuracy of the translation. Brevity penalties are used to account for translations that are less wordy than the reference. The following formula is used to get the BLEU score:

$$\mathrm{BLEU} = \mathrm{bp} \times \exp\left(\sum p_n\right)$$

In BLEU scoring, there's a mechanism called Brevity Penalty (bp) that adjusts scores to handle cases where translations are shorter than the reference ones. bp is calculated by comparing the total word count of reference translations to that of the machine-generated one. Precision of n-grams (p_n) is another aspect, evaluating how many n-grams match between the reference and machine-generated translations, divided by the total n-grams in the machine-generated translation.

4.5 EPERIMENTATION RESULT:

4.5.1 IndicBART:

- **ROGUE SCORE:**

	rouge-1	rouge-2	rouge-l
recall	0.526316	0.247619	0.515789
precision	0.574713	0.254902	0.563218
f-measure	0.549451	0.251208	0.538462

Table 2:Rogue Score (IndicBART)

- **BLEU SCORE AND RELATED METRICS:**

Metric	Value
BLEU Score	0.133607
1-gram-precision	0.404000
2-gram-precision	0.192700
3-gram-precision	0.102958
4-gram-precision	0.059781
brevity penalty	0.903037
length ratio	0.907447
translation length	101292.000000
reference length	111623.000000

Table 3:BLEU metrics (IndicBART)

4.5.2 mT5:

- **ROGUE SCORE:**

	rouge-1	rouge-2	rouge-l
recall	0.378947	0.095238	0.357895
precision	0.404494	0.097087	0.382022
f-measure	0.391304	0.096154	0.369565

Table 4:Rogue Score (mT5)

- **BLEU SCORE AND RELATED METRICS:**

Metric	Value
BLEU Score	0.089154
1-gram-precision	0.313743
2-gram-precision	0.131226
3-gram-precision	0.061919
4-gram-precision	0.032408
brevity penalty	0.935139
length ratio	0.937155
translation length	104608.000000
reference length	111623.000000

Table 5:BlEU Metrics (mT5)

4.5.3 Graph Comparison:

- **ROGUE SCORE GRAPH (IndicBART):**

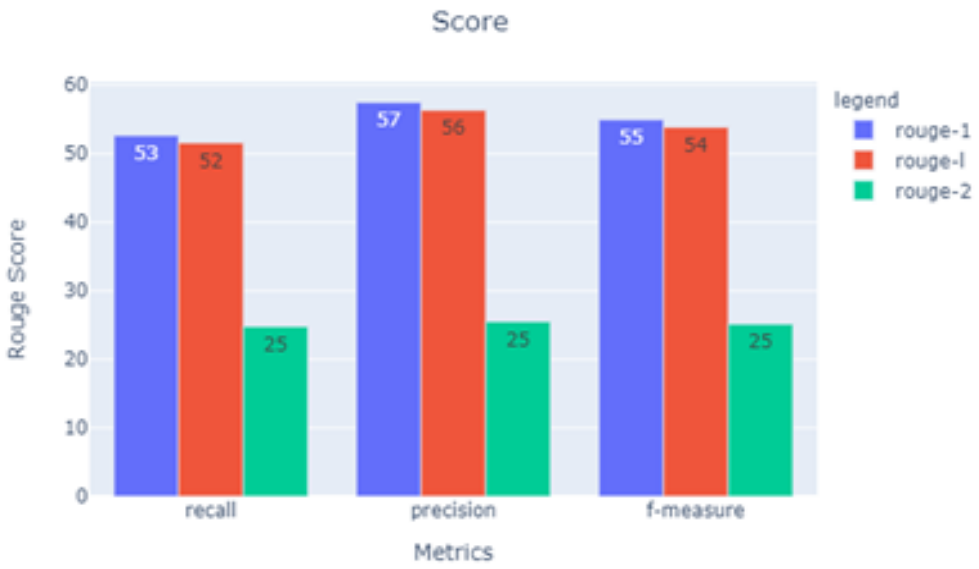


Figure 5: IndicBART Rogue Scores

- **ROGUE SCORE GRAPH (mT5):**

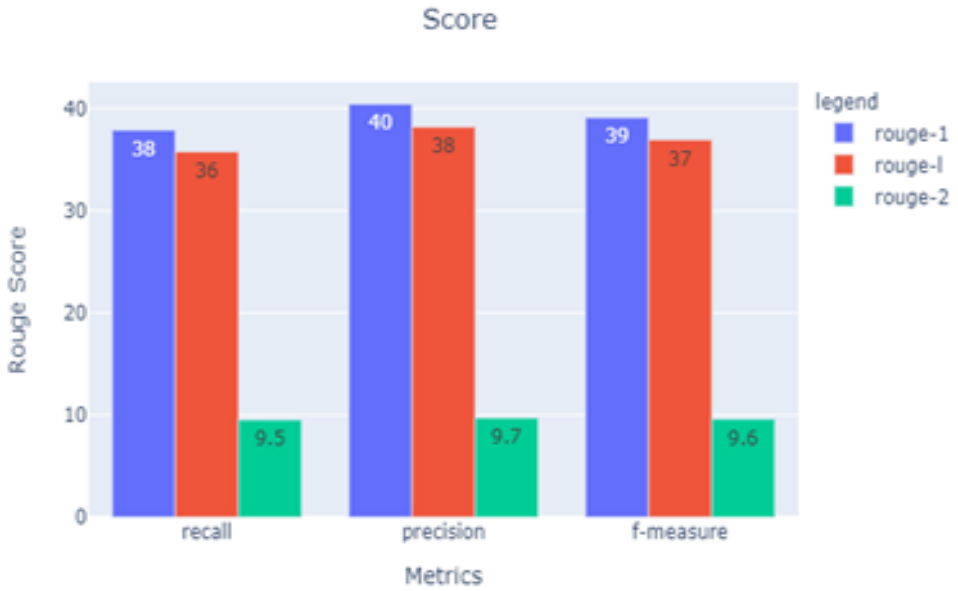


Figure 6: mT5 Rogue Scores

This table outlines a comparison between two methods, one extractive and the other abstractive, used for text summarization. It examines their performance using varied Hindi datasets and techniques/models. The comparison provides valuable insights into how these approaches fare across different evaluation metrics and techniques/models, offering a comprehensive overview of their effectiveness in text summarization.

		EXTRACTIVE APPROACH				**ABSTRACTIVE APPROACH**	
DATASET		Literary novels in Hindi				Hindi news Corpus	
TECHNIQUES AND MODELS		Document-vector embedding	TextRank	LexRank	LSA (Latent Semantic Analysis)	Indic-BART	mT5
rouge-1	f-score	0.27	0.25	0.19	0.21	0.54	0.39
rouge-2		0.1	0.07	0.06	0.06	0.25	0.09
rouge-l		0.17	0.13	0.13	0.13	0.53	0.36
rouge-1	precision	0.17	0.14	0.15	0.14	0.57	0.40
rouge-2		0.06	0.05	0.04	0.04	0.25	0.09
rouge-l		0.11	0.09	0.09	0.09	0.56	0.38
rouge-1	recall	0.48	0.47	0.42	0.43	0.52	0.37
rouge-2		0.21	0.19	0.13	0.13	0.24	0.09
rouge-l		0.33	0.3	0.25	0.25	0.51	0.35

Table 6: Comparison of methods

CHAPTER 5
CONCLUSION

The field of news headline generation, which is a subset of automatic text summarizing, is vital to the process of reducing long texts into brief summaries, which in turn saves readers time. However, existing text summarization efforts have primarily focused on the English language, leaving a gap in available tools for Indian languages. This study aimed to address this gap by utilizing datasets from prominent Hindi news outlets. We conducted fine tuning on two pre trained language models, IndicBART and mT5, and comprehensively evaluated their performance using metrics like BLEU and ROGUE scores. IndicBART achieved a ROUGE-1 score of 54.9%, while mT5 scored 39.13% on the text summarization task. Additionally, the BLEU scores for IndicBART and mT5 were 0.13 and 0.08, respectively. The superior performance of IndicBART over mT5 underscores its potential to significantly improve automated summarization systems, thereby enhancing information retrieval and understanding for the Hindi-speaking community.

By affixing a bidirectional encoder to the autoregressive decoder to construct a denoising autoencoder architecture, BART performs better than the T5 model. Its auto-regressive decoder makes it an excellent choice for language generation tasks. The ROUGE-1 score demonstrates how well both models perform on challenging text summarizing tasks. The dataset is used as an input and includes a variety of headline and news pairs in Hindi. Most of the time, the headline accurately conveys the news's context. While a machine would need more knowledge, people might be able to understand the gap.

5.1 Future Work:

The future of abstractive Hindi text summarization involves improving natural language generation techniques, refining strategies for handling complex grammar and semantics, and customizing summaries to meet diverse user needs and domains. Furthermore, efforts will focus on addressing linguistic and cultural nuances specific to the Hindi language.

- This project mainly focuses on the mono-lingual(Hindi) for news summarization. The model can adapt to summarize news articles in multiple languages, thereby making it more versatile and globally applicable.
- Experimentation with more advanced transformer architectures like Longformer, Reformer, or BigBird to handle longer sequences more effectively and capture richer contextual information.

- Develop techniques to enable real-time summarization of live news feeds or streaming content, ensuring timely and relevant summaries for users.

REFERENCES

[1] Rani, R., Lobiyal, D.K. Document vector embedding based extractive text summarization system for Hindi and English text. Appl Intell 52, 9353–9372 (2022).

[2] Pingali, P., Jagarlamudi, J., and Varma, V. A dictionary based approach with query expansion to cross language query based multi-document summarization: Experiments in telugu-english. National Workshop on Artificial Intelligence, 2006.

[3] Naidu, R., Bharti, S. K., Babu, K. S., and Mohapatra, R. K. Text summarization with automatic keyword extraction in telugu e-newspapers. In Smart Computing and Informatics, pages 555–564. Springer, 2018.

[4] D. N. S. Y Madhavee Latha. Multi-document abstractive text summarization through semantic similarity matrix for telugu language. International Journal of Advanced Science and Technology, 29(1):513–521, 2020.

[5] Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document transformer. arXiv:2004.05150,2020.

[6] Manjari, K. U. Extractive summarization of telugu documents using textrank algorithm. In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pages 678–683. IEEE, 2020.

[7] Bhosale, S., Joshi, D., Bhise, V., and Deshmukh, R. Marathi e-newspaper text summarization using automatic keyword extraction technique. International Journal of Advance Engineering and Research Development, 5(3):789–792, 2018.

[8] Renjith, S., and Sony, P. An automatic text summarization for malayalam using sentence extraction. In proceedings of 27th IRF International Conference, 2015.

[9] Burney, A, Sami, B., Mahmood, N., Abbas, Z., and Rizwan, K. Urdu text summarizer using sentence weight algorithm for word processors. International Journal of Computer Applications, 46(19):38–43, 2012.

[10] Cachola, I., Lo, K., Cohan, A., and Weld, D. TLDR: Extreme summarization of scientific documents. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4766–4777, Online, Nov. 2020. Association for Computational Linguistics.

[11] Cao, Z., Li, W., Li, S., and Wei, F. Retrieve, rerank and rewrite: Soft template based neural summarization. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 152–161, 2018.

[12] See, A., Liu, P. J., and Manning, C. D. Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[13] Cohan, A., Dernoncourt, F., Kim, D. S., et al. A discourse-aware attention model for abstractive summarization of long documents. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[14] Narayan, S., Cohen, S. B., and Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1797–1807, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[15] Sarkar, K. Bengali text summarization by sentence extraction. preprint arXiv:1201.2240, 2012.

[16] Rush, A. M., Chopra, S., and Weston, J. A neural attention model for abstractive sentence summarization. CoRR, abs/1509.00685, 2015.

[17] Damodar, R., Ramineni, A. and Konda, R. Telugu text summarization. International Research Journal of Modernization in Engineering Technology and Science, India, 2021.

[18] Dou, Z.Y., Liu, P., Hayashi, H., Jiang, Z. and Neubig, G. GSum: A general framework for guided neural abstractive summarization. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4830–4842, Online, June 2021. Association for Computational Linguistics.

[19] Fabbri, A. R., Wu, X., Iyer, S., and Diab, M. Multi-perspective abstractive answer summarization. arXiv preprint arXiv:2104.08536, 2021.

[20] Li, C., Xu, W., Li, S., and Gao, S. Guiding generation for abstractive text summarization based on key information guide network. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 55–60, 2018.

[21] Yasunaga, M., Kasai, J., Zhang, R., Fabbri, A. R., Li, I., Friedman, D., and Radev, D. R. Scisummnet: A large, annotated corpus and content-impact models for scientific paper summarization with citation networks. In Proceedings of the AAAI conference on artificial intelligence, volume 33, pages 7386–7393, 2019.

[22] Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L. and Ahmed, A. Big bird: Transformers for longer sequences, 2021.

[23] Graff, D., Kong, J., Chen, K. and Maeda, K. English gigaword. Linguistic Data Consortium, Philadelphia,4(1):34, 2003.

[24] Over, P., and Yen, J. An introduction to duc-2004. National Institute of Standards and Technology, 2004.

[25] Thaokar, C. and Malik, L. Test model for summarizing hindi text using extraction method. In 2013 IEEE Conference on Information & Communication Technologies, pages 1138–1143. IEEE, 2013

[26] Hanumanthappa, M., Narayana Swamy, M. and Jyothi, N. Automatic keyword extraction from Dravidian language. International Journal of Innovative Science Engineering and Technology, 1(8):87–92, 2014.

[27] Pattnaik, S., and Nayak, A. K. A simple and efficient text summarization model for odia text documents. Indian journal of computer science and engineering, 11, 2020.

[28] Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom. P. Teaching machines to read and comprehend. Advances in

neural information processing systems, 28:1693–1701, 2015.

[29] Meng, R., Thaker, K., Zhang, L., Dong, Y., Yuan, X., Wang, T. Bringing structure into summaries: a faceted summarization dataset for long scientific documents. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 1080–1089, Online, Aug. 2021. Association for Computational Linguistics.

[30] Kallimani, J. S., Srinivasa, K. et al. Information retrieval by text summarization for an indian regional language. In Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering (NLPKE-2010), pages 1–4. IEEE, 2010.

[31] Kallimani, J. S., Srinivasa, K. et al. Information extraction by an abstractive text summarization for an Indian regional language. In 2011 7th International Conference on Natural Language Processing and Knowledge Engineering, pages 319–322. IEEE, 2011.

[32] Khanam, M. H., and Sravani, S. Text summarization for telugu document. IOSR Journal of Computer Engineering (IOSR-JCE), 18(6):25–28, 2016.

[33] Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., and Okumura, M. Controlling output length in neural encoder-decoders. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1328–1338, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[34] Rush, A. M., Chopra, S., and Weston, J. A neural attention model for abstractive sentence summarization. CoRR, abs/1509.00685, 2015.

[35] Nallapati, R., Xiang, B., and Zhou, B. Sequence-to-sequence rnns for text summarization. CoRR, abs/1602.06023, 2016.

[36] Rao, R., Sharma, S. & Malik, N. Automatic text summarization using transformer-based language models. Int J Syst Assur Eng Manag (2024).

[37] Singh, M., and Yadav, V., "Abstractive Text Summarization Using Attention-based Stacked LSTM," pp. 236–241, Oct. 2022, doi: 10.1109/CCICT56684.2022.00052.

[38] Shah, A., Zanzmera, D., and Mehta, K., "Deep Learning based Automatic Hindi Text Summarization," Proc. - 6th Int. Conf. Comput. Methodol. Commun. ICCMC 2022, pp. 1455–1461, 2022, doi: 10.1109/ICCMC53470.2022.9753735.