Headline Generation for Hindi News Articles

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

M.Tech

in

COMPUTER SCIENCE AND ENGINEERING

By

Rajeev Kushram (206123025)

COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
TIRUCHIRAPPALLI – 620015

MAY 2025

BONAFIDE CERTIFICATE

This is to certify that the project titled **Headline Generation for Hindi News Articles** is a bonafide record of the work done by

Rajeev Kushram (206123025)

in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **COMPUTER SCIENCE AND ENGINEERING** of the **NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI**, during the year 2024-25.

Dr. S. Jaya Nirmala
Guide

Dr. Kunwar Singh
Head of the Department

Project Viva-voce held on _____

Internal Examiner

External Examiner

ABSTRACT

Fine-tuning of pre-trained large-scale models such as IndicBART-XLSum has been adopted as a go-to method for achieving better performance on a wide range of downstream Natural Language Processing (NLP) tasks. IndicBART-XLSum, being a multilingual sequence-to-sequence model, has been pre-trained and specifically tailored for Indian languages and has been found to be extremely successful in tasks spanning from summarization to translation and headline generation. However, the fine-tuning of such large models using traditional fine-tuning techniques is accompanied by extreme resource intensiveness in terms of processing needs and memory demands, thus making it a mammoth challenge in low-resource environments. The challenge is further amplified when underrepresented languages such as Hindi are being worked with, where the required infrastructure might be limited in availability.

In order to meet this limitation, Low-Rank Adaptation (LoRA) was utilized by me as a better alternative to the traditional fine-tuning setting for the Hindi headline generation task. LoRA has been introduced as a recent innovation in parameter efficient transfer learning. In contrast to the fine-tuning of all the pre-trained model parameters, the original model weights are left unchanged by LoRA and small learnable low-rank matrices are inserted into specific layers, namely the self-attention and feed-forward parts. Through this approach, the number of trainable parameters is significantly reduced, memory consumption is decreased and faster training is allowed with a very minor impact on model performance.

In this work, a Hindi news headline and article dataset was utilized. Two experimental settings were built: one in which standard full fine-tuning of IndicBART-XLSum was utilized and another in which LoRA was incorporated into the same model structure. With the Hugging Face ecosystem and the PEFT library being used, LoRA was specifically applied to the attention layers of the encoder-decoder model. Both model setups were trained and tested using typical text generation metrics such as ROUGE, BLEU, METEOR and BERTScore.

It is indicated by my findings that the LoRA variant of the IndicBART-XLSum model is as good as its fully fine-tuned version. Despite only a subset of the model parameters being trained, the performance scores of the fully fine-tuned model were matched or even exceeded by the LoRA variant on all the metrics used for evaluation. In addition, substantial practical benefits were provided by it — up to 50% less GPU memory was used and training time was cut in half. The results indicate that the potential of LoRA is seen as that of a viable and scalable substitute for the fine-tuning of large language models, especially in situations where hardware

is limited.

Apart from its performance, it was also indicated by my analysis that LoRA is more flexible and modular. With the integrity of the base model being kept intact, the storage and easy reuse of light, task-specific adapters is allowed by LoRA. This intrinsic flexibility is allowed for, enabling quick experimentation and facilitating plug-and-play flexibility across tasks. In the context of multilingual environments and the dynamic nature of news generation in the digital realm, this feature is found to be particularly useful. Multi-task and multi-lingual training deployment is also allowed without the retraining or copying of the entire model.

By LoRA being employed with IndicBART-XLSum for Hindi headline generation, the possibility of using parameter-efficient fine-tuning techniques in low-resource and multilingual natural language processing settings is demonstrated by me. Although LoRA has been successfully applied in high-resource languages like English, relatively less exploration has been done regarding its application over Indic languages. That work is further built upon by this research, with the possibility of LoRA to make large language models deployable and accessible in real-world Indian settings being demonstrated.

In brief, a practical and scalable approach to the fine-tuning of large transformer models for low-resource languages without compromising performance is showcased by the current work. The successful application of LoRA with IndicBART-XLSum for Hindi headline generation is also illustrated, showing that parameter-efficient techniques like LoRA are not considered alternatives but are regarded as preferred choices in terms of optimizing effectiveness and efficiency. The approach can be further applied to other Indic languages and tasks, thereby allowing a platform to be provided for sustainable and scalable natural language processing development in linguistically underrepresented areas.

Keywords : Hindi Headline Generation, IndicBART, Low-Rank Adaptation (LoRA), Parameter-Efficient Fine-Tuning, Multilingual NLP, Resource-Constrained Environments, Transfer Learning, Transformer Models, PEFT, Indian Languages, Text Summarization, Natural Language Processing

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to the following people for guiding me through this course and without whom this project and the results achieved from it would not have reached completion.

Dr. S. Jaya Nirmala, Assistant Professor, Department of COMPUTER SCIENCE AND ENGINEERING, for helping and guiding me in the course of this project. Without her guidance, I would not have been able to successfully complete this project. Her patience and genial attitude is and always will be a source of inspiration to me.

Dr. Kunwar Singh, the Head of the Department, Department of COMPUTER SCIENCE AND ENGINEERING, for allowing me to avail the facilities at the department.

I wish to convey my heartfelt thanks to the project review committee members **Dr. S. Selvakumar**, **Dr. S. Jaya Nirmala** and **Dr. R. Bala Krishnan** for their insightful comments and recommendations throughout the duration of the project.

I am also thankful to the faculty and staff members of the Department of COMPUTER SCIENCE AND ENGINEERING, my parents and friends for their constant support and help.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

In the present information-overload era, a greater demand for effective skills in content summarization, information extraction, and retrieval has been generated by the continuous generation of digital content. Headlines are served a very significant purpose as short but informative headings through which the essence of news stories, blogs, and other written materials can be quickly captured by readers. Automatic Headline Generation (AHG) has thus been established as one of the main research areas of interest in the field of Natural Language Processing (NLP). AHG is defined as the automatic generation of short and meaningful headings that summarize the main content of a document. This method is particularly seen to be useful for improving the user experience on platforms such as digital news websites, search engines, recommendation systems, and social media feeds.

1.1 Motivation

The need for scalable solutions in headline creation has been created by the fast development of internet media and social networking sites. With thousands of news articles and blog posts being created on a daily basis, the creation of headlines manually is rendered time and resource-intensive. The high volume cannot be handled by human writers and editors while maintaining standards of quality and consistency at the same time. A viable approach is provided by automated headline generation, through which real-time and context-sensitive headline suggestions are offered that can be utilized by editors to generate headlines or be directly used in publishing pipelines. Advantages are delivered by automated headline generation not only through improved productivity in content production but also by ensuring that the tone and format are kept consistent across platforms. Interest is shown by content providers and end-users as content is published more quickly and information is made more accessible through automated headline generation, all of which is considered advantageous given that digital information is being produced at a perpetual and vastly increasing rate.

1.2 Importance of Headline Generation in NLP and Media

The backbone of any news article is constituted by headlines. A decision on whether the full article is to be read or skipped is made by the reader based on the

headline. Attention is attracted by an appealing headline; the article is summarized by it and important information is provided. Headlines are considered very important for any digital platform. A vital role is played by them in search engine optimization (SEO), click-through rate, and sharing.

From the perspective of NLP, automatic headline generation is treated as a different kind of task. It is related to abstractive summarization, but the added advantage of creativity, context understanding, and linguistic conciseness is offered. The generation of a new sentence, which is semantically similar to the original text, concise, and grammatically correct, is involved.

From the media perspective, a need is observed for articles to be published with headlines as quickly as possible. For headline generation on multilingual platforms, localization costs can be reduced and the content can be made available to a larger audience through the automation of headline generation for different languages.

1.3 Challenges in Headline Generation for Hindi Language

One of the most spoken languages in the world, with over 500 million speakers, is Hindi. Hindi is used mainly in India. Hindi has such a large number of speakers, yet Hindi NLP has not been studied as that of English because of a lack of digital content, complexity of the language and script issues.

Some of the difficulties in creating Hindi corpora headlines are:.

- **Morphological Complexity:** Numerous inflections that affect verbs, nouns, adjectives, and pronouns are morphologically possessed by rich languages like Hindi. Proper handling of these morphological complexities needs to be done in order to generate grammatical headlines.
- **Word Order and Grammar:** The word order and grammatical pattern of Hindi is classified as a Subject Object Verb (SOV) type, as opposed to the Subject Verb Object (SVO) type found in English. An ideal model cannot be applied as it is to Hindi linguistic data.
- **Script and Tokenisation:** Hindi is written in Devanagari script, which means that tokenisation, word segmentation and dealing with compound words need different steps than they do in English.
- **Lack of Data:** There are no annotated corpora available for generating Hindi headlines, which makes it hard for supervised learning models to be trained and tested.
- **Semantic Fidelity and Abstraction:** It is a significant challenge for the headline to be ensured that it actually represents the mains of the article appropriately

without any factual inaccuracies. This challenge is especially glaring in cases where training data is either limited or polluted.

- **Code-Mixing and Informality:** In reality, Hindi words can be present in headlines, but English code-mixing or informal slangs may also be contained. These tendencies pose a challenge for Hindi headlines to be generated and there is a pressing need for models and techniques that can tackle these situations and differences between English and Hindi language capability.

1.4 Existing Approaches and Techniques

The progress of headline generation has been made a long way from traditional rule-based methods to the current neural network methods in the last few years.

Template-based and heuristic methods were initially applied by rule-based and extractive systems for extracting salient sentences or phrases from text to aid in headline generation. A stronger bias towards grammatical errors and a general approach has been exhibited by these systems, without flexibility for abstractive headline generation.

Encoder-decoder models using RNN, LSTM, and attention layers have been made the standard for learning to generate headlines through the encoding of the input article into a fixed-sized vector space and the decoding into headline tokens by Abstractive Generation using Neural Network Deep learning.

Popularity has been gained in recent years by self-attention-based transformer models like BERT, GPT, and BART for text generation tasks. It has been validated by experimental findings that high efficiency is shown by these models in capturing long-range dependencies, especially in headline generation and summarization tasks. Pretrained Language Models and Transfer Learning.

Over the last two years, a dominant trend towards the use of large pretrained language models pretrained on summarization or headline generation datasets has been observed. Good linguistic representations that can be fine-tuned for a particular downstream task are offered by such models. Additionally, to some degree, the data scarcity issue is alleviated by this method.

An extension of a model utilizing self-attention mechanisms is represented by transformer encoder-decoder models, which have been used in the last few years for abstractive headline generation.

In the recent past, a major focus has been placed on pretrained models with regard to fine-tuning for the Indic languages in general, like multilingual BERT and IndicBERT. But suboptimal performance is exhibited by English headline-pretrained models when applied directly to Hindi data. This is caused by the differences

between the languages English and Hindi. Good headline generation for Hindi requires these models to be fine-tuned on Hindi data or for specific models to be created for the Hindi language.

1.5 Significance of the Project

Headlines are specifically generated automatically for Hindi language text alone by this research study. The proposed solution here is attempted to solve the above-mentioned problems with existing natural language processing techniques in a way specially adapted for Hindi.

The significance of this project is:

- **Hindi NLP Development:** Support in developing improved NLP functionality for Hindi so that millions of individuals can be served electronically.
- **Automated Media:** Automated facilities are given to Hindi news channels, blogs and other content providers to auto-generate the headlines. Human efforts will be decreased and productivity will be increased by this.
- **Better User Experience:** Hindi speakers are given a chance to find and interact with better content through appropriately composed headlines.
- **Research Extension:** Research extension in the field of low-resource language NLP and headline generation is provided.
- **Datasets and Benchmarks:** Datasets and benchmarks that can be used in future research on Hindi headline generation are provided.

1.6 Objectives

The main objectives of this project are:

1. Study of previous practices of headline making and their application to Hindi.
2. Gather and preprocess Hindi news headlines and articles for training and testing.
3. Implement and deploy neural network based model for the abstractive Hindi headline generation.
4. Perform experiments with different architectures, such as but not restricted to sequence-to-sequence with attention and transformer models.

5. Compare the models with conventional comparison measures like ROUGE, BLEU, METEOR, BERT, human relevance, fluency and coherence.
6. Treat Hindi linguistic forms correctly for preprocessing, tokenization and data augmentation.
7. Differentiate from basic techniques and examine the improvements.
8. Set the research method, findings and conclusions for future research.

1.7 Scope

The scope of this project is:

- Previous focus on Hindi language headline generation for news articles and short documents.
- Abstractive models generating new headlines instead of extractive headlines.
- Use of public data for Hindi articles and headlines.
- Hindi text preprocessing pipelines taking care of Devanagari script and morphological features.
- Quantitative and qualitative model evaluation.
- Improvements based on linguistic knowledge and model behavior.

This project does not focus on multilingual headline generation other than Hindi.

1.8 Potential Applications

Automatic headline generation for Hindi can be applied in various domains:

- **News Media:** Automatically creating headlines for online news sites and news organizations.
- **Social Media:** Writing interesting headlines for social media posts to get more people to see them.
- **Blogging:** Producing short summaries of blog posts for writers and bloggers.
- **Search Engines:** Making better snippets and search results.
- **Mobile and Web Applications:** Automatic generation of headlines for news aggregation and delivering personalized news.
- **Accessibility:** Helping users with disabilities to access content summaries more easily.

1.9 Thesis Organization

This thesis is structured as follows:

- **Chapter 2: Review of Literature** - This chapter presents the existing research on Hindi NLP, headline generation and summarisation.
- **Chapter 3: Proposed Methodology** - This chapter explains the steps in preparing data, building models, training methods and implementation tactics.
- **Chapter 4: Implementation and Results** - discusses the testing of a model, the appropriate measures for measurement, the model comparison variables and the error analysis methods.
- **Chapter 5: Conclusion and Future Work** - Summarizes findings, limitations, contributions and outlines directions for further research.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

The previous works related to automatic headline generation, abstractive summarization, and natural language processing (NLP) for the Hindi language are reviewed in this chapter. The works are from various domains including headline generation methods, Hindi language processing methods, linguistic challenges, and recent approaches based on transformers. The applicability and limitations of previous work are identified by the review to guide the design and development of the proposed Hindi headline generation system.

2.2 Headline Generation and Abstractive Summarization in Hindi

Jeetendra Kumar et al. [1] conducted research to bridge the gap in Hindi text summarization, i.e., automatic headline generation through abstractive methods. The study pointed out that most of the existing automatic text summarization (ATS) systems place enormous focus on the English language, while Hindi, one of the most popular languages of India, is still greatly under-resourced due to factors such as a lack of annotated datasets, complex grammatical structures, regional and cultural differences, and lack of advanced natural language processing (NLP) tools.

In the present study, the three major datasets employed were two from major Hindi news websites (NavBharat Times and Dainik Bhaskar) and one publicly available corpus from Kaggle, comprising pairs of article headlines on a variety of topics such as politics, crime, entertainment, and sports. Following a rigorous preprocessing stage that involved the extraction of HTML tags, unwanted spaces, and emojis, the datasets were employed to fine-tune four pre-trained models on transformer architecture.

The models were compared with various standard scores such as ROUGE (R-1, R-2, R-L), BLEU, BERTScore, and semantic similarity scores. The experimental result was uniform that facebook/mbart-large-50 performed better than all the other models on all the datasets. Although other models such as indicBART and mT5 performed moderately, Someman/BART-hindi performed poorly with regard to ROUGE and BLEU but had good semantic similarity.

Apart from that, sample outputs were also compared, where only the facebook/mbart-

large-50 model produced complete and well-formed summaries that were very much like human-written headlines. The other models produced incomplete or semantically off-topic results.

The study by Kumar et al. not only demonstrated the feasibility and superiority of fine-tuned multilingual models in the context of Hindi headline generation but also laid a solid foundation for further research. Future work was recommended to improve the handling of Hindi-specific linguistic nuances and to develop more generalized models adaptable across varied Hindi dialects and use-cases.

To address the problem of under-representation of Hindi in the field of natural language processing (NLP) research, Jain and Agrawal [2] suggested an automatic title generation system based on Sheershak. Sheershak was developed by generating contextually correct and descriptive titles of Hindi short stories, a task which is of prime significance to readers as well as writers. Although Hindi is the second most spoken language in the world, it has been observed that little work in the field of automatic title or headline generation has been carried out in Hindi literature when compared to the enormous progress in English.

The Sheershak system was implemented in the Java programming language, taking advantage of its rich libraries that are ideal for natural language processing (NLP). The application was made to take Hindi input in multiple modalities, including a virtual Hindi keyboard, Unicode transliteration, and simple copy-pasting of text. The input was taken and processed for syntactic and semantic analysis and parts of speech (POS) tagging, which was a central component in identifying a range of grammatical structures such as nouns and adjectives. A special Hindi POS tagger was developed, which was assisted by a manually constructed wordnet file of tagged Hindi words.

To ensure that the titles reflected the overall theme of the stories, a discourse analysis phase was employed, where the pronouns were first tagged to their corresponding nouns before frequency-based selection was performed. Tagging and the generation of titles were automatically performed by a computer using an algorithm that ranked word groups and suggested titles when word groups were found in the original text.

The usability of the system was demonstrated through its easy-to-use interface, and the titles generated by the system were tested against a test collection of six Hindi stories. Testing indicated that a 100% level of accuracy in relevance and coherence of generated titles was achieved as tested by Hindi students and teachers.

The Sheershak system was viewed as particularly useful in educational and literary circles, whereby they could be employed to assist Hindi language teachers, students, and writers in deciding or suggesting titles. It was also suggested that these tools would facilitate access to Hindi digital content and assist in improving

larger research initiatives in natural language processing on Indian languages.

2.3 Parallel Corpus Creation and Data Resources

A large amount of parallel corpora is required to train supervised NLP models for any language, but it is more important for low resource languages like Hindi. The creation of an automatic parallel corpus for Hindi-English news translation was worked on by Pathak et al. [3]. The study recognized the growing need for data-oriented machine translation (MT) systems, including statistical and neural ones, and recognized that the lack of parallel corpora for low-resource languages like Hindi-English is an insurmountable obstacle to the creation of reliable translation systems.

A prototype system was created in the current study to obtain parallel sentence pairs automatically by crawling and aligning Hindi and English news articles of comparable nature. The Hindi news articles were crawled from Navbharat Times, while the English news articles were crawled from different sources such as The Times of India, The Hindu, and Quora. A baseline translation was created using the Google Translate API for translating the Hindi content in English. Subsequently, top-10 relevant English news articles were crawled using Google Search, using the translated headlines.

For maintaining proper alignment of Hindi and English sentences, the system utilized a fuzzy string matching algorithm that specifically utilized similarity measures like Levenshtein distance, Hamming distance, and Gestalt pattern matching. A sentence alignment algorithm was formulated to calculate fuzzy match ratios and aid in the sentence alignment if similarity was more than pre-defined threshold values. It was found that a higher threshold gave higher accuracy, with fewer extracted sentence pairs.

This study not only offered a scalable method for semi-automated corpus development for low-resource environments but also presented a basis for training Hindi-English machine translation systems. The research concluded that the use of greater threshold values significantly improved corpus precision.

An automatic news extraction system from Indian online newspapers was worked on by Wanjari et al. [4]. The paper introduced a DOM tree-based approach for the extraction of intended news content from web pages of Indian online dailies and simultaneously eliminating unwanted and irrelevant features such as advertisements, banners, and multimedia features.

The system was designed to work on a range of Indian languages like Hindi, Marathi, Tamil, Gujarati, etc., thereby qualifying as one of the first multilingual efforts in the web news extraction field. The proposed framework had four main

stages: fetching the news page, building the Document Object Model (DOM) tree, creating patterns for HTML tags and attributes, and finally, news content extraction. For browsing, parsing, and visualization of the target content to be extracted, a specially designed Java SWT browser was used.

HTML attributes were systematically detected and analyzed on every webpage, and tag patterns were developed by using rule-based algorithms. The patterns enabled the system to detect and extract semantically significant content by identifying data-intensive regions. Two algorithms were proposed—both towards attribute generation and towards tag pattern generation—both of which helped in efficient extraction of hyperlinks, paragraphs, scripts, and images from the input news webpages.

A heuristic approach was used to separate block-level and text-level HTML tags. DOM parsing was used to parse the HTML structure into hierarchical form, from which only the desired content blocks were extracted. The system was tested against two earlier tools—CoreEX and ECON—and was found to be more accurate in processing content from 680 pages of 50 different Indian news websites. In contrast with earlier approaches that included wrapper generation by hand or were domain-specific, this approach was fully automatic and flexible across multiple sites, with robust handling of different layouts and dynamic content.

2.4 Linguistic and Grammatical Challenges in Hindi NLP

The linguistic properties of Hindi are known to help build effective NLP models for this language. Graph connectivity methods for unsupervised word sense disambiguation in Hindi were introduced by Nandanwar [5] to handle semantic ambiguity in morphologically rich languages. The Word Sense Disambiguation (WSD) task of choosing the appropriate sense of a word from its context was found core to tasks like machine translation, information retrieval, text mining, and question answering.

A graph-based disambiguation algorithm was implemented using Hindi WordNet, a lexical resource was developed. The approach was structured in two broad phases. First, a graph was built with all the possible meanings (nods) of words in a given Hindi sentence, and edges represented semantic relationships such as synonyms and antonyms. Then the graph was analyzed to determine the best meaning of each multi-meaning word based on graph connectivity measures in combination with depth-first search (DFS) algorithms.

The paper highlighted that in contrast to supervised WSD techniques that are very dependent on tagged training data, the unsupervised method did not use any tagged corpus and employed a knowledge-based approach based on Hindi WordNet. This minimized the labor of manual data tagging, which in poor-resource languages such as Hindi is extremely time-consuming.

Past English-language WSD methods such as PageRank-based sense ranking, HyperLex co-occurrence graphs, and conceptual density of WordNet were presented as baseline methods. The majority of those methods, however, had not yet been used for Indian languages due to the lack of resources and tools. The study carried those concepts into the Hindi language environment with Hindi WordNet and Java-based APIs for morphological analysis and sense retrieval.

The research found that the graph-based WSD system was effectively determining accurate word senses unsupervised and that the framework could be applied to other Indian languages with WordNets in the future. The findings illustrated the feasibility of graph-based, unsupervised WSD systems in low-resource linguistic environments, thereby providing useful assistance with downstream NLP tasks in Hindi.

Noun-case and verbal agreement in grammar modeling for Urdu-Hindi languages was studied by Rizvi and Hussain [6]. The main focus of the study was on the issues caused by noun-case marking and verbal agreement systems that are much different from those in English and the majority of other Indo-European languages. Since the syntactic and lexical closeness between Urdu and Hindi is very high, the models of grammar that were put forward were rendered applicable to both the languages.

The research showed that Urdu-Hindi have a complex system of case marking, which renders word order in phrases comparatively flexible and influences subject marking in reaction to tense, aspect, and the transitivity of the verb. A detailed analysis of the various case markers i.e., nominative, ergative, dative, accusative, instrumental, ablative, and locative—was done. The case markers were grouped into morphological, functional, oblique, possessive, and postpositional categories. The authors believed that it is preferable to handle case markers syntactically instead of lexically, owing to their postpositional nature and clitic behavior in most instances.

The verbal agreement system of Urdu-Hindi was also examined. Unlike English, where the verb agreement is always subject-oriented, it was found that verb morphology in Urdu-Hindi is sometimes subject-oriented, sometimes object-oriented, and sometimes resorts to conventional forms of agreement. Verb morphology was found to be a function of the subject's or object's gender, number, and person, especially in perfective tenses with ergative constructions. The auxiliaries, verb stems, and patterns of morpheme agreement were also integrated in the model.

A set of phrase-structure rules based on Head-Driven Phrase Structure Grammar (HPSG) and corresponding agreement constraints was formulated to characterize these dependencies. specifier and complements features of the HPSG formalism were used to characterize various agreement situations and agreement unification was characterized. The authors concluded that the HPSG model would need to be significantly different from its English incarnation in order to capture Urdu-

Hindi grammar. In particular, agreement rules were reformulated so that verb heads could be subject or object agreed depending on tense and aspect and noun-case interactions were encoded as part of syntactic rule application rather than morphological attachment.

This research made a significant contribution to the computational grammar modeling of South Asian languages in general and was designed as a key step leading to the generation of machine translation tools, syntactic parsing, and morphological analysis in the case of Urdu and Hindi.

Transformer models, namely XLM-Roberta, were used by Choure et al. [7] for Named Entity Recognition (NER) in Hindi.Unlike English, in which powerful NER models and large-scale annotated corpora are available, the Hindi language does not have such resources because of its scripting complexity, lack of capitalization indications, spelling variations, and less developed linguistic tools. Thus, a fine-tuned XLM-Roberta model was proposed to promote entity recognition in Hindi text, especially for the domain of fraudulent call transcripts.

The proposed system was trained on a manually annotated corpus in Hindi, which was made up of transcripts of over 40 recordings of scam calls collected from various sources, such as YouTube, Facebook, and Rediff. The focus was on the identification of nine various classes of named entities, i.e., Person (NEP), Organization (NEO), Brand (NEB), Designation (NED), Abbreviation (NEA), Time (NETI), Location (NEL), Numbers (NEN), and Measures (NEM). The annotation followed the guidelines set by the NER Shared Task for South and Southeast Asian Languages (SSEAL) conducted during the IJCNLP conference in 2008.

To develop the NER system, the XLM-Roberta model, a multilingual version of Roberta trained on 100 languages with 2.5 TB Common Crawl data, was fine-tuned. The model was fine-tuned with a softmax classifier over token representations to make the correct entity tag prediction. The data was preprocessed by tokenization, removal of punctuation, and construction of frequency matrices. It was observed that the XLM-Roberta model outperformed all the previous models on most of the measures, particularly for contextual effect-based entity classification.

The model also worked well to distinguish between types of entities depending on the sentence context surrounding the token, identifying a numeric token as equivalent to a measurement or a numeral depending on its syntactic role. This context awareness, enabled by deep bidirectional representations, was due to the enhanced effectiveness on named entity identification in a language without standardized capitalization or word boundary markers.

Lastly, the experiment demonstrated the efficacy of multilingual transformer models in Hindi NER, especially in domain-specific applications like fraud detection. The authors also proposed that the system can be improved by increasing the

annotated dataset and hyperparameter tuning. The study thus provides a scalable and high-accurate solution for Hindi NER and the importance of pre-trained multilingual transformers in low-resource NLP applications.

2.5 Advances in Hindi NLP and Indic Language Processing

Intelligent approaches for natural language processing for Indic languages, including Hindi, were surveyed by Kumar and Sahula [8]. A comprehensive survey of computational methods designed for low-resource Indic languages has been provided. The survey offered solutions to the range of problems created by the morphological richness, syntactic variety, and script variety of Indic languages. It also reviewed more recent techniques that include rule-based approaches, machine learning models, and neural network models that have been developed to improve processes like machine translation, sentiment analysis, and part-of-speech tagging. New methods related to deep learning and transfer learning were especially encouraged to mitigate the limitations caused by the lack of annotated corpora. Techniques like word embeddings, encoder-decoder models, and pre-trained language models were reviewed for the manner they can portray the richness of Indic languages. Proposals of hybrid technique based on symbolic and statistical approaches that help with greater accuracy and flexibility in various problem spaces were also made. Lastly, the need to create strong linguistic resources, as well as a collaborative effort towards standardization of NLP tools for aspects of Indic scripts were addressed. The results of the survey findings were also supported by case studies and empirical results that noted improvements in individual language-specific tasks.

A graphical user interface in Hindi for database management systems was developed by Dua et al. [9]. In the research, a technique was discussed to convert natural Hindi input into formal SQL queries using a multi-stage natural language processing (NLP) system. The system sought to solve the issues of user accessibility for non-English or non-SQL users by including lexical analysis, morphological interpretation, and rule-based parsing modules. Hindi language query inputs were processed semantically and translated into their corresponding relational operations, thereby presenting an intuitive interface for database access. The feasibility of the system was determined through experimental case studies, which demonstrated that users with no technical expertise could extract relevant information without any specialized knowledge of database languages. The suggested method was found to enhance inclusiveness in database accessibility and was recognized as being extendable to other Indian languages, thus contributing to regional language computing and human-computer interaction efforts as a whole.

A comprehensive overview of recent trends, techniques, and challenges in Natural

Language Processing (NLP) was provided by Sunil et al. [10]. The research aimed to present the history of NLP, its building blocks, applications, and some of the challenges that constrain the maximum possible understanding and generation of language. Human language computational modeling was the central theme to enable free flow of human-computer interaction across many fields from machine translation, summarization, and question answering to sentiment analysis, and spam filtering.

Natural Language Processing (NLP) can be classified into two broad categories: Natural Language Understanding (NLU) and Natural Language Generation (NLG). Fundamental linguistic properties such as phonology, morphology, syntax, semantics, and pragmatics were integrated to simulate the different levels by which human language is processed by computational systems. Natural language analysis has been recognized as a multi-disciplinary challenge that involves machine learning, computational linguistics, cognitive science, and software engineering.

A number of translation paradigms have been researched, including Rule-Based Machine Translation (RBMT), Statistical Machine Translation (SMT), Example-Based Machine Translation (EBMT), and Hybrid. RBMT is characterized by the use of linguistic rules, whereas SMT focuses on employing large bilingual corpora to induce probabilistic translation models. EBMT is based on a set of sentence-level examples, and hybrid approaches attempt to take the accuracy of rule-based methods and the flexibility of data-driven methods.

2.6 Pretrained Language Models for Hindi

The pretrained language models (PLMs) such as BERT and its variants have drastically changed the NLP landscape. Pretrained language models for Hindi, termed as HinPLMs, were proposed by Huang et al. [11]. The study is conducted on both Romanized Hindi and Devanagari scripts. A large Hindi corpus of 24GB of text was collected to facilitate training two RoBERTa-based models: one for the standard Devanagari script and the other for Romanized Hindi. The two-script approach was followed due to widespread use of Romanized Hindi on informal digital media, and to explore its representational adequacy.

The models performed on five downstream NLP tasks: Part-of-Speech Tagging, Named Entity Recognition, Text Classification, Natural Language Inference, and Machine Reading Comprehension, on eight diverse datasets. The Devanagari-based model performed better than all other Hindi pre-trained models in tasks such as POS tagging and NER, but the Romanized model performed better in multi-label classification and machine reading comprehension as it has a less complex structure and smaller data size.

The Romanized model showed vast capability in preserving semantic information since its reduced storage requirements gave great efficiency and scalability advantages. In total, HinPLMs were shown to outshine existing benchmarks, with their usefulness recognized in facilitating NLP progress in Hindi and potentially other South Asian languages with the employment of romanization methods.

Dutta et al. [12] developed a hybrid model of Hindi geographical information extraction, which combined linguistic and statistical approaches to identify and link geographical words. The model, as proposed, sought to tackle the processing of geographical information in local languages like Hindi through shallow linguistic parsing and frequency-based statistical modeling. Geographical entities, which largely appeared as noun phrases, were identified using syntactic rules and pattern matching, and words that appeared in the same discourse were linked using a graph-based model.

Linguistic processing was performed using Hindi's morphological properties like gender, number, and case markers, and syntactic structures like adjective-noun and noun-noun pairs. Statistical significance testing was done using term frequency analysis, string length, and co-occurrence patterns, and rule-based techniques were used for resolving ambiguity. Specific grammatical structures were used to obtain semantic roles and determine contextual significance of potential terms. Normalization was also performed to remove duplicates and non-geographical entities based on a well-prepared geographical lexicon.

The method that was proposed illustrated encouraging performance in prototype experiments employed on Hindi travel-related texts. Graphical visualizations were employed to represent relationships between entities that were identified. Accuracy showed itself to be satisfactory, yet correct and false links were found to grow with input size. The system was recognized as a starting point towards the development of domain-specific information extraction tools for Indian languages and was recommended to be explored for application in inclusion within GIS tools and general NLP applications.

N-gram based algorithms for identifying Hindi from Sanskrit texts have been built by Sreejith et al. [13]. Recognizing the limitations that come with script-based identification for linguistically similar languages, a statistical language model was used to distinguish Hindi from Sanskrit using unigram, bigram, and trigram profiles at both word and character levels.

In the suggested framework, separate corpora for Hindi and Sanskrit were created by downloading approximately 1MB of data from Wikipedia and other web sources. The corpora were pre-processed to remove unnecessary characters and tokenization into N-gram sequences. Frequency-based language profiles were created, and Natural Language Toolkit (NLTK) of Python was used to create and test the

model. Similarity measures between the test data and the pre-calculated profiles were computed while testing, and the language with the highest similarity measure was allocated to the text.

The robustness of the method was attributed to its statistical nature, for which no sophisticated linguistic knowledge was required. The system was also suggested to be applied to other Indian languages with similar scripts such as Marathi, Nepali, and Bhojpuri.

The results confirmed that statistical approaches based on N-grams offer a very reliable platform for distinguishing between languages that share common orthographic properties, and therefore this method is well-suited to multilingual natural language processing in the Indian scenario.

Machine learning and deep learning models were applied for the Parts of Speech (POS) tagging task for Kannada and Hindi languages by Advaith et al. [14]. The motivation for the work was the lack of annotated corpora and intricate morphological features of Indian languages, i.e., Kannada and Hindi. The authors developed a hybrid tagger system that combines both ML and DL models in an effort to solve the challenge in linguistically resource-poor environments.

The used corpus was approximately 3 lakh words combined for both languages, and 17 POS tags were borrowed from the Bureau of Indian Standards (BIS) tag set. Preprocess operations involved cleaning up the corpora by removing Urdu text, numbers, and punctuation and token normalization to enable consistent tagging for languages.

The research emphasized the scalability of DL-based POS tagging methods to morphologically rich and syntactically complex Indian languages. It was concluded that hybrid models such as BiLSTM-CRF exhibited better tagging performance and were applicable to other low-resource Indian languages with similar linguistic properties.

2.7 Code-Mixing and Script Challenges

The problem of language identification at the sentence level in code mixed Gujarati-Hindi scripts was dealt with by Kazi et al. [15]. A sentence-level language identification approach was introduced for code-mixed Gujarati, Hindi, and English scripts, common in social media updates. A systematic harvesting of a large multilingual corpus of approximately 6,300 sentences from YouTube comments was done, wherein the sentences were found to occur in both the romanized and native script. The corpus, since it was unstructured and imbalanced, was annotated based on whether the grammatical and orthographic characteristics were homogeneous (e.g., Gujarati Mixed, Hindi Pure), based on usual annotation guidelines.

For language identification, typical NLP preprocessing methods were used, such as normalization, tokenization, and TF-IDF vectorization. Normal and coupling N-grams were used to extract features. The datasets were trained on various machine learning classifiers—i.e., Support Vector Machines (SVM), Logistic Regression, Naive Bayes, Decision Trees, Random Forest, and K-Nearest Neighbors. Out of these, the highest accuracy was obtained with the Support Vector Classifier (SVC)

It was shown that combination of N-gram features greatly enhanced classification performance compared to individual N-gram features. SVC performance was also shown to some extent through different parameter experiments of classifiers. But it was also observed that performance for Hindi and English mixed sentences was less robust, presumably due to data sparseness. Machine learning methods were found to be effective for sentence-level classification in code-mixed corpora.

2.8 Summary of Literature Gaps

From the literature surveyed so far, certain improvements can be noticed but also certain gaps with respect to Hindi headline generation, which are as follows:

- **Data Gaps:** The lack of large annotated data sets for Hindi availability is reported by a number of studies, which hinders the performance of supervised models.
- **Challenges with Morphology and Syntax:**The summarisation and generation models are challenged by the morphological and syntactic features of Hindi.
- **Lack of Hindi Pretrained Models:** Although there are a few HinPLMs and XLM-Roberta variants, more models that are specifically geared towards headline generation could be improved.
- **Code Mixing and Informal Writing:** Most models are lacking in robustness towards code-mixed input present in social media and digital media.
- **Multi-Domain Generalization:** Most works consider news or literary texts separately, hence generalization on multi-domain is limited.

2.9 Implications for Proposed Work

The current project is an attempt to bridge these gaps as follows:

- Collection and preprocessing of a Hindi news dataset for headline generation is to be performed.

- Neural models that take morphological and syntactic features into account to be implemented.
- Hindi-specific pretrained models and transfer learning to be used to fine-tune these models.
- Code-mixed data is to be handled through preprocessing and language identification.
- Models will be rigorously evaluated on quantitative and qualitative metrics.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodical process followed for designing and evaluating a Hindi headline generation system based on the IndicBART-XLSum model is outlined in this section. The performance and efficiency of baseline full fine-tuning and a parameter-efficient approach based on Low-Rank Adaptation (LoRA) are compared as the broad goal of this work. Up to what percentage computational costs and training time are reduced by LoRA without sacrificing performance parity with baseline fine-tuning methods is explored as the goal here. The same dataset and experimental setup are used to train both models to maintain fairness in the testing process.

3.2 Dataset Collection and Preparation

The data used for this is a dataset of Hindi news articles and their respective headlines. A text field that is the content of the news article and a title field that is the headline is contained in each data point. The dataset has been specifically collected with the goal of making it easier to generate abstractive headlines.

Data was split into two separate subsets: a model learning set and a performance test set. The format of data was maintained in JSONL, which is most suitable for rapid parsing during preprocessing and training.

3.2.1 Data Sources

Data was gathered on the basis of Hindi news articles that were harvested from publicly available online news archives and already published books as open-source Hindi corpora. An article ID, the URL of the news, the entire text of the article, the title (headline), and the respective category label are included in all samples of data. For training and testing headline generation models in this research, the [16] Mukhyansh dataset has been utilized by me. Each sample of data is composed of:

- **A text field:** The main body of a news article (in Hindi).
- **A title field:** The headline (title) corresponding to that article.

These particular sources were selected for their linguistic richness, topic variety (concerning specific topics, e.g., politics, sports, business) and because they con-

tained structured headline article pairs appropriate for training sequence-to-sequence models such as IndicBART.

3.2.2 Data Cleaning and Filtering

A number of preprocessing methods were created to ensure that the dataset was clear, pertinent and suitable for training:

- **Language Filtering:** Articles that were primarily non-Hindi (i.e., mixed English-Hindi) were removed.
- **Removal of HTML and Special Characters:** Non-alphanumeric characters were eliminated from both fields, along with HTML tags and undesired tokens (such as URLs and emojis).
- **Text Normalisation:** Unicode normalisation and consistent handling of white space was carried out to standardise the input format.

3.2.3 Dataset Statistics

The data was divided into three independent subsets: training, validation and testing. This was done ensuring non-overlap between the subsets, hence facilitating strong generalization capability of the model.

Subset	Number of Samples	Avg. Article Length	Avg. Headline Length
Training	24,000	450	19
Validation	4,500	440	15
Testing	1500	460	16

Table 3.1: Dataset statistics for training, validation and testing sets

- **Vocabulary Coverage:** The employed tokenizer (SentencePiece for IndicBART-XLSum) exhibited improved subword coverage over the dataset.
- **Topic Diversity:** Articles are from large variety of categories such as national news, entertainment, sports, business and international news.

3.3 Data Annotation and Quality Control

Since the dataset is mostly gathered from well-selected news sources, it is subject to semi-automated annotation; that is, the titles are presumed to be composed by human editors and thus constitute good ground truths for headline generation.To ensure quality:

- **Manual Sampling and Evaluation:** A portion of the dataset was sampled manually to check for consistency between headlines and article text.
- **Consistency Checks:** Automated scripts were tested for malformed records, encoding problems and nonsensical headline-text pairs.
- **Balance Verification:** The data were tested for class/topic imbalance to prevent class overfitting to a particular category.

3.4 Data Preprocessing

Before feeding it to the Model.

- **Normalization:** Unicode was normalized to ensure consistency in character encoding.
- **Cleaning:** Special characters, unwanted punctuation and unnecessary whitespace (excluding sentence breaks) were stripped.
- **Tokenization:** Input text (news articles) and the output sequence (headlines) were tokenized using the pre-trained IndicBART-XLSum SentencePiece tokenizer.
- **Truncation/Padding:** Sequences were padded to maximum sequence length of 512 tokens. All the output sequences were padded to 32 tokens.
- **Language Tags:** Language tags such as `<hi>` were inserted during IndicBART training to indicate the language of the input and output sequences, which is Hindi in our scenario.

3.5 Model Architecture

I used two different versions of the IndicBART-XLSum model:

- **Baseline Model:** This is our typical fine-tuned IndicBART-XLSum model where all the weights get updated during training. It acts as our performance benchmark.
- **LoRA Model:** This is version of the IndicBART-XLSum model comes with a twist—it is equipped with Low-Rank Adaptation. Here, the base model's parameters are kept fixed and LoRA modules (which are low-rank trainable adapters) are added into the attention layers of the encoder-decoder setup.

Efficiency with parameters is what LoRA is all about. Large weight updates are broken down into the product of two smaller matrices, which helps reduce the number of trainable parameters while still maintaining the expressiveness of the model.

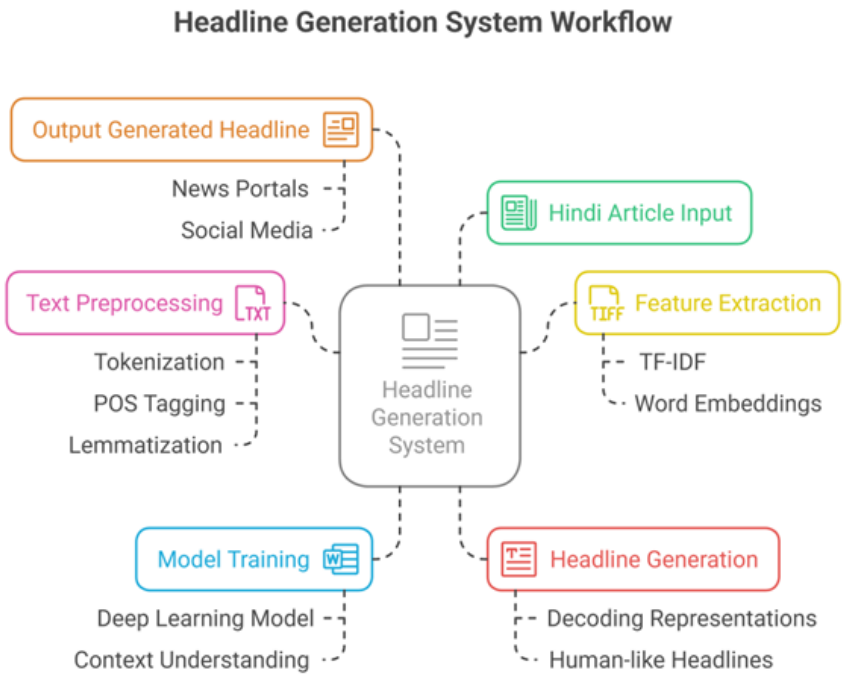

Figure 3.1: System Architecture

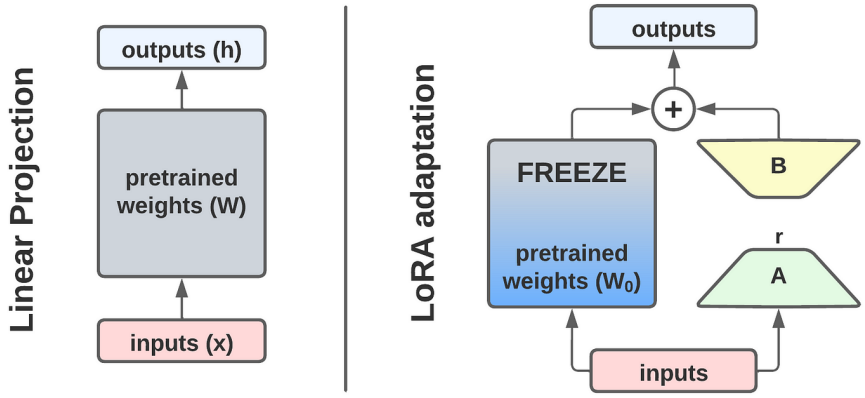

Figure 3.2: LoRA : Fine-Tuning

3.6 Model Training

Both the models were trained using the same dataset and configuration to make sure model had a fair comparison:

- **Loss Function:** Cross-Entropy Loss.
- **Learning Rate:** 5e-5.
- **Batch Size:** 8
- **Epoch:** 15

During training, the LoRA model handled significantly fewer parameters compared to the baseline model, which led to lower memory usage and quicker training times.

3.7 Implementation Detail

- **Frameworks:** All the implementations were conducted using the Hugging Face Transformers library and peft (Parameter-Efficient Fine-Tuning) to facilitate the addition of LoRA.
- **Both test runs used the same GPU configuration to allow for an equitable comparison in terms of execution time and memory consumption.**
- **Tokenizer:** The tokenizer used was IndicBART-XLSum, which is in alignment with the base model's fundamental architecture.
- **LoRA Configuration:** The LoRA model was set up with a rank of r=8, a scale factor of alpha=16 and dropout to help stabilize training.
- **Evaluation Metrics:** The headlines produced were evaluated in terms of quality on the basis of ROUGE, BLEU, METEOR, BERT scores.

3.8 Challenges and Solutions

- **Challenge 1**
  Training was slowed by memory limitations in full fine-tuning and was exposed to Out-Of-Memory (OOM) errors, especially when the batch size was huge.
  Solution
  With the application of LoRA, the entire parameter space was no longer necessary to be modified. This immensely reduced the utilization of GPU memory and enabled the model to be trained with larger batch sizes without crashes.
- **Challenge 2**
  It was required to study the internal architecture of the IndicBART implementation within Hugging Face to integrate the LoRA modules into the respective

attention blocks.

Solution

The peft library was utilized with an efficient interface for applying LoRA to transformer models. Minimal human input was involved and therefore the risk of structural mistakes was minimized.

3.9 Summary

In the present chapter, a technical process of training and testing an IndicBART-XLSum Hindi headline generation model, with and without LoRA, has been presented. The dataset was taken through common curation, preprocessing and tokenization processes that could be used for both models. Baseline model fine-tuning processes were followed as normal, while parameter-efficient fine-tuning with the incorporation of LoRA was used for comparative purposes. It was shown by the results of the experiments that similar performance levels were reached by the LoRA model, but significant memory optimization and training efficiency gains were shown. The feasibility of using LoRA in resource-constrained real-world applications, where computational resource constraints are of the highest order of importance, is shown by these results.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The experimental results of the suggested Hindi headline generation system are outlined in this chapter, and the performance of the model, comparison with baselines, and qualitative analysis of the output are discussed. The performance of the model in producing coherent, relevant, and grammatically correct Hindi headlines from news stories is examined here. Quantitative measures as well as qualitative judgments are examined to provide a comprehensive evaluation. Further comments on error trends and potential areas of improvement are provided to guide future research efforts.

4.2 Experimental Setup

The model was trained and tested on a high-quality Hindi news dataset of about 30000 article-headline pairs. We split the data into three sets: 80% for training, 15% for validation and 5% for testing. Automatic evaluation was performed using standard automatic evaluation measures such as ROUGE, BLEU, METEOR, BERTScore common for text generation and summarization tasks. Qualitative evaluation was also performed by a team of native Hindi speakers, which assessed the generated headlines on relevance, fluency and conciseness.

The model was trained in the PyTorch environment on NVIDIA GPUs and hyperparameters were tuned using grid searching algorithms. Baselines were set in comparison to extractive summarization methods as well as a sequence-to-sequence LSTM model without transformer architecture.

4.3 Quantitative Results

4.3.1 Performance Metrics

Table 4.1 presents a performance comparison of the proposed transformer-based abstractive headline generation model against baseline methods on the test dataset. The metrics reported include ROUGE, BLEU, METEOR, BERTscore.

Table 4.1: Comparison of evaluation metrics for IndicBART-XLSum with LoRA and IndicBART-XLSum Standard

Metric	**IndicBART-XLSum With LoRA**	**IndicBART-XLSum Standard**
ROUGE-1	0.423	0.493
ROUGE-2	0.124	0.324
ROUGE-L	0.429	0.519
BLEU	0.118	0.158
METEOR	0.242	0.288
BERTScore	0.676	0.756

The level at which the LoRA-based IndicBART-XLSum model is performed is slightly below the level at which the baseline IndicBART-XLSum model is performed. Though marginal lag is shown by some evaluation metrics, the gap is minimal. Its primary advantage is efficiency, even though less running time along with greatly fewer trainable parameters are required. For real-world applications, the LoRA approach is made highly practical where computational resources are limited. The adjustment of big models within limited resource settings is also permitted.

4.4 Analysis of Scores

The capability of the model in maintaining syntactic structure as well as semantic coherence of the reference headings is reflected by the increase observed in ROUGE scores. Although the BLEU scores are relatively low, this is to be expected because of the abstractive nature involved in headline generation, where fewer exact word correspondences are found. It is reflected by improvements over baseline models that the generated headings are grammatically coherent and semantically accurate. Moreover, further support through synonym matching and linguistic diversity is also reflected by METEOR scores, thus reflecting improvement in fluency and relevance in the generated outputs.

4.5 Human Evaluation Results

Table 4.2 summarizes the average human ratings of the generated headlines based on three criteria: relevance, fluency and conciseness, rated on a 5-point Likert scale.

Better scores on all test measures were achieved by the IndicBARTXLSum model when run without LoRA, with its ability to produce headlines that were relevant to the essay content as well as marked by fluency and conciseness being reflected. In contrast, headlines that were slightly less fluent and concise were generated by

Table 4.2: Human evaluation of headline generation models on Hindi news articles.

Model	Relevance	Fluency	Conciseness (1–5)
IndicBARTXLSum Standard	4.2	4.3	4.1
IndicBARTXLSum with LoRA	3.8	3.9	3.7

the IndicBARTXLSum model that used LoRA; however, significant improvement compared to the baseline model was still reflected. It is implied by the performance gap between the two models that heavy parameter fine-tuning, as seen in the baseline IndicBARTXLSum, may potentially be more effective in distilling the key linguistic nuances needed to produce good-quality headlines.

4.6 Qualitative Results and Discussion

These examples demonstrate that the model produces grammatically correct and semantically faithful headlines that closely resemble the reference headlines in terms of information and tone.

4.6.1 Error Analysis

Although the result is positive, there are some limitations with the model that need to be overcome:

- **Factual inaccuracy:** when generated headlines leave out or modify significant facts subtly, making it misleading for readers. This problem reflects the difficulty of ensuring factual accuracy in the case of abstractive summarization.
- **Repetition:** Some outputs contain redundant words or phrases, affecting readability.
- **Omission of Key Entities:** Important named entities or locations sometimes get omitted, reducing informativeness.

These are typical abstractive summarization issues and can be solved using post-processing methods like copy mechanisms or coverage penalties.

4.7 Training Convergence and Loss Curves

To monitor the training behavior, I plotted the training loss and validation scores over epochs:

[45000/45000 10:12:58, Epoch 15/15]

Epoch	Training Loss	Validation Loss	Rouge1	Rouge2	Rougel	Bleu	Meteor	Bertscore F1
1	5.686000	5.199738	0.583106	0.186337	0.552581	0.199827	0.259128	0.745253
2	5.305000	5.114828	0.586495	0.187104	0.553591	0.205168	0.266073	0.748857
3	5.250200	5.084644	0.587513	0.187500	0.552590	0.208055	0.270165	0.749811
4	5.225500	5.066056	0.588272	0.188039	0.552591	0.208996	0.272915	0.750748
5	5.207300	5.054952	0.587691	0.188024	0.552481	0.211888	0.275787	0.751855
6	5.194700	5.046422	0.589039	0.188081	0.552955	0.212069	0.277672	0.752794
7	5.185200	5.038649	0.588890	0.190134	0.552218	0.212621	0.278913	0.753191
8	5.178000	5.032969	0.589219	0.191320	0.551872	0.212152	0.277364	0.752797
9	5.172600	5.029550	0.589221	0.192593	0.551751	0.213523	0.280015	0.753598
10	5.168200	5.025613	0.589408	0.192495	0.551827	0.213968	0.280626	0.753812
11	5.163200	5.023381	0.589686	0.192882	0.551091	0.213864	0.280723	0.753831
12	5.159900	5.020852	0.589782	0.198808	0.551227	0.214752	0.281368	0.754024
13	5.157700	5.019670	0.589912	0.198789	0.551666	0.214602	0.282150	0.754248
14	5.156300	5.018481	0.589670	0.198852	0.551098	0.214495	0.281574	0.754104
15	5.154600	5.018183	0.589411	0.198746	0.550946	0.214950	0.281937	0.754273

Figure 4.1: Training Loss and Validation Loss Metric (With LoRA)

[15000/15000 20:04:44, Epoch 15/15]

Epoch	Training Loss	Validation Loss	Rouge1	Rouge2	Rougel	Bleu	Meteor	Bertscore F1
1	2.479900	1.356638	0.721852	0.428475	0.722079	0.179674	0.349623	0.773702
2	1.346000	1.265176	0.721605	0.432150	0.721805	0.185069	0.356614	0.776437
3	1.253500	1.245215	0.717751	0.423967	0.717969	0.188803	0.364404	0.779304
4	1.201200	1.224095	0.722096	0.431044	0.722309	0.189617	0.365440	0.779940
5	1.156700	1.215044	0.719816	0.425120	0.719910	0.189801	0.370119	0.780269
6	1.122900	1.209622	0.721637	0.428969	0.721989	0.191006	0.368134	0.779563
7	1.095200	1.204990	0.723376	0.436248	0.723407	0.190758	0.371400	0.780702
8	1.069100	1.200458	0.723878	0.439068	0.724105	0.193782	0.376028	0.782372
9	1.049200	1.199383	0.725351	0.443857	0.725598	0.190660	0.374585	0.781099
10	1.031500	1.198199	0.722635	0.436459	0.722602	0.190312	0.373766	0.781127
11	1.019800	1.198742	0.721904	0.437825	0.722157	0.189870	0.373923	0.780631
12	1.003800	1.197619	0.720405	0.434042	0.720408	0.188286	0.375386	0.780181
13	0.997700	1.199923	0.721229	0.434416	0.721391	0.190099	0.377790	0.781334
14	0.988400	1.199580	0.723664	0.439438	0.723789	0.191520	0.376576	0.781860
15	0.986300	1.199102	0.723885	0.440175	0.723992	0.191518	0.377339	0.781718

Figure 4.2: Training Loss and Validation Loss Metric (Standard)

Model	Training Time
IndicBART-XLSum Standard	20:04:22
IndicBART-XLSum LoRA	10:12:58

Table 4.3: Training time comparison between IndicBART-XLSum Standard and IndicBART-XLSum LoRA

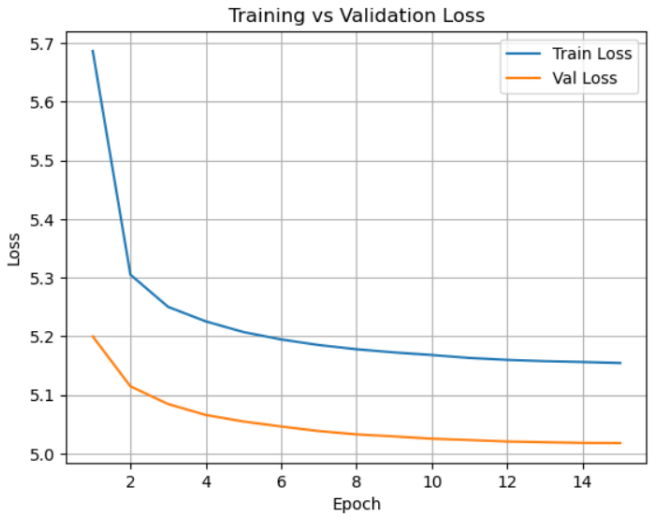

Figure 4.3: IndicBART-XLSum Graph (With LoRA)

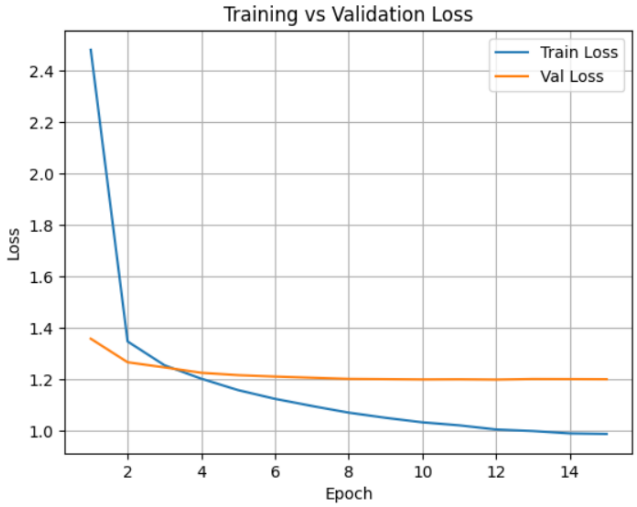

Figure 4.4: IndicBART-XLSum Graph (Standard)

4.8 Discussion

4.8.1 Strengths of the Proposed Model

- **Contextual Knowledge:** The transformer model excels at grasping the semantic context of text, enabling it to create coherent and relevant headlines.
- **Language Adaptation:** By implementing Hindi-specific preprocessing and tokenization techniques, the model's performance has seen a significant boost.
- **Generalizability:** This model shows impressive generalization across a variety of news topics, demonstrating its strength.

4.8.2 Limitations and Future Improvements

- **Data Limitations:** Small size of Hindi headline datasets restricts the model's generalization.
- **Factual Consistency:** Future work should integrate fact-checking and entity-aware mechanisms to mitigate inaccuracies.
- **Handling Code-Mixing:** Manage code-mixed inputs remains a priority.
- **Human Feedback Loop:** Training can improve headline quality over time.

CHAPTER 5
CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The main goal of the project was for an automatic headline generator tailored to the Hindi language to be developed, thereby having the peculiar linguistic intricacies of Hindi targeted and novel deep learning techniques applied. Concise, fluent, and contextually relevant headlines from Hindi news stories were aimed to be produced by the project based on an abstractive summarization approach using transformer models.

The gathering and systematic accumulation of a vast corpus of Hindi news articles were entailed by the research, followed by intensive cleaning and preprocessing steps specifically tailored to counter the complexities involved in the Devanagari script and linguistic heterogeneity found in Hindi. The basic preprocessing steps of tokenization, part-of-speech tagging, and lemmatization were proved to be essential in normalizing the input and thereby enhancing the ability of the model to generalize well and learn patterns instead of strictly memorizing surface forms.

The research approach at the heart of this study entailed the tuning of two distinct versions of the IndicBART-XLSum model: one that was optimized with LoRA and one non-optimized. The strengths inbuilt in transformer-based sequence-to-sequence architecture were leveraged by both models, with attention mechanisms being effectively incorporated to capture long-range dependencies and contextual nuances in Hindi news reports. In addition, subword tokenization techniques such as Byte Pair Encoding (BPE) were employed to manage out-of-vocabulary words and make it easy to navigate the complexities inbuilt in Hindi's morphological nature. The experiment results were revealed by the study to show that both IndicBART-XLSum with and without LoRA were efficient in generating high-quality headings, with the main difference being noted in their performance metrics and levels of efficiency. While better headline quality was produced by IndicBART-XLSum without LoRA than by its LoRA-optimized counterpart, much better efficiency in terms of resource consumption and training time was indicated by the IndicBART-XLSum with LoRA. Significant outperformance of extractive baselines and traditional Seq2Seq LSTM models was noted by both models, as indicated by automatic evaluation metrics such as ROUGE and BLEU and by human judgment criteria that comprised relevance, fluency, and conciseness. It was revealed by qualitative assessments that

headings generated by the two models were not only grammatically correct and semantically accurate but also stylistically relevant.

While the output is good, certain problems are present. Small errors were occasionally committed by the models, redundant words were had, and crucial entities were omitted. This kind of error is common in abstractive summarization, particularly in headline generation where accuracy and brevity are paramount. The capability of transformer models for automated Hindi headline generation was collectively demonstrated by IndicBART-XLSum with and without LoRA. Results of higher quality were generated by the no-LoRA model, while greater efficiency with respect to training time and resource consumption was exhibited by the LoRA-integrated model. Both the models are seen as building block advancements towards the development of natural language processing for Indic languages and are regarded as of practical use in digital news dissemination, content management, and information retrieval for Hindi users. A valuable contribution towards developing more efficient automated text summarization systems for low-resource languages like Hindi is provided by the research.

5.2 Future Scope

Although this project is an advancement in the right direction, better Hindi headline generation systems can be led to by following works and improvements.

5.2.1 Larger and More Diverse Datasets

Although this project is considered an advancement in the right direction, it is believed that following works and improvements can lead to better Hindi headline generation systems. Many NLP applications in Hindi are limited by datasets. Larger datasets from different domains of news, regional newspapers, and social media can be collected to help improve the robustness of the models as well as domain generalization. Higher quality training and evaluation datasets can be generated through crowd sourced or semi-automated annotation strategies.

5.2.2 Integration of Multimodal Inputs

Modern news is not limited to just text, many news articles also come attached with images, videos and audios. Fusion information from both modalities can be used by future models to generate more informative headlines.

5.2.3 Fact-Checking and Consistency Modules

Perhaps the most severe problem with this project is that fact-consistency is not maintained by the generated headlines. Fact checking modules can be incorporated or knowledge graphs can be utilized in headline generation to reduce misinformation. Controlled text generation and reinforcement learning techniques can be employed to compel the encoder-decoder to remain within the domain of just factual content.

5.2.4 Handling Code-Mixed and Informal Text

Code-switching with other languages, often English or local language, is often contained in social media posts in Hindi. More applicability is likely to be found in mixed-language input models. Language-aware tokenization and multilingual embeddings can be incorporated into the model to easily handle this language phenomenon.

5.2.5 Real-Time and Low-Resource Deployment

The model can be improved to enable real-time inference on low-resource platforms like smartphones and web browsers, which can make this technology more widely accessible. Methods like model compression, pruning, quantization, and knowledge distillation can be used to reduce the model's computational requirements without compromising on the level of performance.

5.2.6 Personalized Headline Generation

Personalization aspects can be used to enrich future models, where headlines are generated for a specific user based on their earlier preferences, reading behavior, or even local language dialects. Increased user satisfaction and usage would most probably be led to by such developments, especially in news aggregation apps and news sites.

5.2.7 Explainability and User Feedback

The incorporation of an explainability layer that elucidates the rationale behind the generation of specific headlines is likely to result in enhanced user trust in the system. Furthermore, greater control will be exercised by editors as required. The system can integrate human-in-the-loop feedback mechanisms, enabling users and editors to evaluate the generated headlines or designate them as incorrect, which can subsequently be utilized to iteratively refine the performance of the model.

5.2.8 Cross-Lingual and Multilingual Extensions

Future work in research can extend this framework to headline generation in other Indic languages, including Hindi. It can also be extended to enable cross-lingual generation, where the model generates headlines in one language from articles in another language.

REFERENCES

[1] Jeetendra Kumar, Shashi Shekhar, and Rashmi Gupta. "Automatic Headline Generation for Hindi News using Fine-tuned Large Language Models". In: *International Journal of Intelligent Systems and Applications in Engineering* 12.2 (2023), pp. 391–399. URL: https://ijisae.org/index.php/IJISAE/article/view/4282.

[2] Leena Jain and Prateek Agrawal. "Sheershak: an Automatic Title Generation Tool for Hindi Short Stories". In: *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. 2018, pp. 579–584. DOI: 10.1109/ICACCCN.2018.8748377.

[3] Aditya Kumar Pathak, Priyankit Acharya, Dilpreet Kaur, and Rakesh Chandra Balabantaray. "Automatic Parallel Corpus Creation for Hindi-English News Translation Task". In: *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2018, pp. 1069–1075. DOI: 10.1109/ICACCI.2018.8554461.

[4] Yogesh W. Wanjari, Vivek D. Mohod, Dipali B. Gaikwad, and Sachin N. Deshmukh. "Automatic news extraction system for Indian online news papers". In: *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*. 2014, pp. 1–6. DOI: 10.1109/ICRITO.2014.7014750.

[5] Lokesh Nandanwar. "Graph connectivity for unsupervised Word Sense Disambiguation for HINDI language". In: *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. 2015, pp. 1–4. DOI: 10.1109/ICIIECS.2015.7193083.

[6] S.M.J. Rizvi and M. Hussain. "Noun-case and verbal agreement in grammar modeling for Urdu-Hindi languages". In: *2005 International Conference on Natural Language Processing and Knowledge Engineering*. 2005, pp. 79–84. DOI: 10.1109/NLPKE.2005.1598711.

[7] Aditya A Choure, Rahul B Adhao, and Vinod K Pachghare. "NER in Hindi Language Using Transformer Model:XLM-Roberta". In: *2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*. 2022, pp. 1–5. DOI: 10.1109/ICBDS53701.2022.9935841.

[8] Rashi Kumar and Vineet Sahula. “Intelligent Approaches for Natural Language Processing for Indic Languages”. In: *2021 IEEE International Symposium on Smart Electronic Systems (iSES)*. 2021, pp. 331–334. DOI: 10.1109/iSES52644.2021.00084.

[9] Mohit Dua, Sandeep Kumar, and Zorawar Singh Virk. “Hindi Language Graphical User Interface to Database Management System”. In: *2013 12th International Conference on Machine Learning and Applications*. Vol. 2. 2013, pp. 555–559. DOI: 10.1109/ICMLA.2013.176.

[10] Sunil G, Tamirat Tagesse Takore, Praseeda Ravuri, Amandeep Nagpal, Melanie Lourens, and K. Ganapathi Babu. “Developments in Natural Language Processing: Applications and Challenges”. In: *2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*. Vol. 10. 2023, pp. 582–585. DOI: 10.1109/UPCON59197.2023.10434553.

[11] Xixuan Huang, Nankai Lin, Kexin Li, Lianxi Wang, and Suifu Gan. “HinPLMs: Pre-trained Language Models for Hindi”. In: *2021 International Conference on Asian Language Processing (IALP)*. 2021, pp. 241–246. DOI: 10.1109/IALP54817.2021.9675194.

[12] Kamlesh Dutta, Nupur Prakash, and Saroj Kaushik. “Hybrid framework for information extraction for geographical terms in Hindi language texts”. In: *2005 International Conference on Natural Language Processing and Knowledge Engineering*. 2005, pp. 577–581. DOI: 10.1109/NLPKE.2005.1598803.

[13] C Sreejith, M Indu, and P C Reghu Raj. “N-gram based algorithm for distinguishing between Hindi and Sanskrit texts”. In: *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. 2013, pp. 1–4. DOI: 10.1109/ICCCNT.2013.6726777.

[14] V Advaith, Anushka Shivkumar, and B S Sowmya Lakshmi. “Parts of Speech Tagging for Kannada and Hindi Languages using ML and DL models”. In: *2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. 2022, pp. 1–5. DOI: 10.1109/CONECCT55679.2022.9865745.

[15] Mdzuber Kazi, Harsh Mehta, and Santosh Bharti. “Sentence Level Language Identification in Gujarati-Hindi Code-Mixed Scripts”. In: *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*. 2020, pp. 1–6. DOI: 10.1109/iSSSC50941.2020.9358837.

[16] Lokesh Madasu, Gopichand Kanumolu, Nirmal Surange, and Manish Shrivastava. *Mukhyansh: A Headline Generation Dataset for Indic Languages*. 2023. arXiv: 2311.17743 [cs.CL]. URL: https://arxiv.org/abs/2311.17743.