International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Automatic Headline Generation for Hindi News using Fine-tuned Large Language Models

Jeetendra Kumar*[1], Shashi Shekhar[2], Rashmi Gupta[3]

Submitted: 16/09/2023 **Revised**: 17/11/2023 **Accepted**: 28/11/2023

Abstract: Generating news headlines is one use case for automated text summarization. With only a few text lines, it creates a summary of the longer content, cutting down on reading time. Text summarization is a very challenging task and it is very difficult to generate summaries as a human being. Most summary tools available in the market primarily concentrate on summarizing English material, resulting in a scarcity of summarizers for other languages. In this study, we used two datasets gathered from Dainik Bhaskar, NavBharat Times, and one publically available dataset to fine-tune four pre-trained language models: Someman/bart-hindi, facebook/mbart-large-50, indicBART, and mT5. To conduct a full performance assessment, a variety of evaluation metrics are utilized. Our comprehensive examination consistently reveals that the facebook/mbart-large-50 model exhibits superior performance compared to other models in terms of these metrics. This highlights its potential to enhance automated summarization systems and facilitate enhanced content retrieval and comprehension within the Hindi-speaking community.

Keywords: *Text Summarization, Fine-tuning, BART, indicBART, mT5*

1. Introduction

Automatic text summarization has various applications, one of which is the production of news headlines by automated processes. Text summarizing is a methodology that involves condensing large bodies of text into shorter versions while retaining crucial information. "Automatic text summarization(ATS) generates summaries containing important sentences and includes all important relevant information from the original document" [1]. The process of summarizing extensive texts has the effect of reducing the amount of time required for reading by condensing the content into a smaller number of text lines. The act of summarizing material is a significant challenge, particularly for human beings who are tasked with this endeavor. The technique of generating news headlines automatically involves the utilization of algorithms and natural language processing (NLP) techniques to create a novel headline for a given text source. The objective of automatic headline generation is to succinctly and informatively summarize the textual content, while also offering prospective readers a brief overview of the material.[2]. The majority of current summarizers are designed to summarize literature written in the English. The scarcity of natural language summarizers in languages other than English is a notable issue. Hindi, an indigenous language of India, possesses a larger repertoire of alphabets and holds the distinction of being the most widely spoken language in the country. Hindi serves as the primary language spoken by a significant majority of the population residing in several regions, including Ranchi, Madhya Pradesh, Chhattisgarh, Uttar Pradesh, Himachal Pradesh, Kolkata, Delhi, and Bihar.[3]. There are several factors contributing to the limited progress in the field of Hindi text summarization. These include the scarcity of annotated Hindi text data, the intricate grammar structure of the language, diverse writing styles, regional variations leading to multiple variants, challenges posed by ambiguity and polysemy, the presence of culture-specific vocabulary, and the restricted availability of natural language processing (NLP) technologies[4].

Summarization methods are divided into three categories[5] i.e. Extractive[6]–[8], Abstractive[9], [10], and Hybrid Text summarization[11]. Extractive summarization has historically been a significant approach employed for the purpose of summary. The algorithm employed in this process specifically chooses significant sentences to comprise the summary. Numerous writers have put forth diverse methodologies for extracting text summarization, employing sentence ranking as a fundamental approach. The provided summaries shown a lack of resemblance to summaries generated by real beings. Following the emergence of Seq2Seq model[12] which utilize an encoder-decoder design, abstractive method of text summarization has been introduced as a method capable of generating summaries that resemble those produced by humans. Transformers, an evolved iteration of Seq2Seq networks, serve as a foundation for several sophisticated language models utilized in the field of summarization. Numerous sophisticated and intricate models have

[1]*GLA University, Mathura, Uttar Pradesh, India*
ORCID: 0000-0002-5705-6369
[2]*GLA University, Mathura, Uttar Pradesh, India*
ORCID: 0000-0001-8824-1447
[3]*Atal Bihari Vajpayee University, Bilaspur, Chhattisgarh, India*
ORCID: 0000-0001-6853-9844
** Corresponding author. E-mail: jeetendragupta85@gmail.com*

undergone training on extensive datasets, rendering them capable of generating summaries even on limited datasets. Transfer learning and fine-tuning are intimately interconnected.

We have used abstractive text summarization in the proposed work for generating headlines from Hindi news. The primary objective of this work was to investigate the field of text summarization specifically in relation to Hindi news headlines, a subject that has received less attention in previous research. We conducted fine-tuning on four language models by utilizing datasets sourced from reputable origins. Subsequently, we evaluated the efficacy of these models by employing diverse measures.

2. Literature Review

Sufficient efforts have been made to summarize text in English and other languages; nonetheless, there is a notable dearth of research on Hindi text summary. The Hindi language has significant dissimilarities from both English and French languages, primarily attributable to its intricate syntax and the presence of *matras* inside its script. Pre-processing the text in the Hindi language is a significant challenge. The predominant approach for Hindi text summarizing has primarily involved the use of extractive summarization techniques. In this series in 2017, Vipul et al.[13] proposed graph based text summarization technique. Their proposed approach for summarizing Hindi literature involved the utilization of the Particle Swarm Optimization (PSO) algorithm to examine the semantic graph created by subject-object-verb (SOV) triples. The classifier, which has been trained using PSO, produces a semantic sub-graph that results in a document summary that is both efficient and concise. In 2021, Manju et al.[14] proposed extractive text summarization method. Their proposed approach involves creating a semantic graph for Hindi text by utilizing the Hindi Wordnet ontology. This allows for the establishment of connections between phrases. They utilize fourteen graph theoretical methods to evaluate sentences and determine their ranking based on semantic scores. The methodology was evaluated using Tourism and Health datasets, surpassing the performance of TextRank in the health field and demonstrating similar outcomes in tourism. The correlation study demonstrated a strong link among the majority of graphical metrics, confirming the effectiveness of our suggested approach. In 2022, Arti et al.[15], proposed a text summarization method. They suggested a unique approach for Automatic Text Summarization (ATS) in Hindi utilizes the Real Coded Genetic Algorithm (RCGA) to analyze health data. The methodology encompasses several stages, including preprocessing, feature extraction, processing, sentence rating, and summary production. These stages are thoroughly tested using a wide range of feature sets. The evaluation, using the ROUGE measure, identifies the most effective combinations of features. The RCGA algorithm is then used to calculate the weights of these features for extractive summarization. This approach achieved a significant 65% reduction in summary length compared to other existing methods. Dhankhar et al.[16] proposed an extractive Hindi text summarization method. Their study used nine textual criteria and a variety of mathematical combinations to score sentences. A comprehensive examination was performed on a dataset including 2000 documents obtained from the Forum of Information Retrieval Evaluation (FIRE) conference. The assessment was conducted using the ROUGE evaluation measures, taking into account a retention rate of 40%. In 2023, Shailendra et al. [17] proposed BPSO based multi-document Hindi text summarization. The BPSO-IGA method, as proposed by authors integrates the Binary Particle Swarm Optimization (BPSO) technique with an enhanced genetic algorithm (IGA) in order to produce coherent summaries from a collection of various documents. When examined on the FIRE 2011 Hindi dataset, this technique demonstrates higher performance compared to six well-known summarization algorithms in terms of precision, recall, F-measure, cohesion, and non-redundancy.

There has been a limited amount of study conducted on the utilization of abstractive text summarizing approaches. In a study conducted by Bhargava et al.[18], authors introduced the utilization of generative adversarial networks for the purpose of multilingual text summarization. The results obtained from their research demonstrated a level of performance that was comparable to existing methods. The research conducted by the authors employs a notably reduced amount of data for the purpose of training the model. The attention-based stacking LSTM model was proposed by Singh et al. [19]. The approach employed by the researchers resulted in the production of succinct summaries. A BLEU score of 0.91 was attained for the test set. In 2022, Shah et al. [20] introduced a text summarizing approach that utilizes deep learning techniques. LSTM neural networks and word embedding were employed in their study. The utilization of word embedding facilitates the transformation of textual data into vector representations, while LSTM cells serve as the fundamental structures for both encoders and decoders inside Seq2Seq networks. After conducting training on a dataset consisting of around 100,000 data points, the researchers were able to attain a ROUGE score of 67.5% and an f-score of 58%. Table 1 presents a concise overview of recent research endeavors conducted in the field of Hindi text summarization.

Table 1. Recent research work on Hindi Text Summarization

Research Work	*Summarization Approach*	*Type of Summarization*	*Data set*	*Method Used*	*Performance measures*
Arti et al. [15]	Extractive	Document Summarization	Hindi Health Data (HHD) corpus Size:- 234 pages of MSWord document	Real coded genetic Algorithm	Rouge_1- 79% Rouge_2- 66%
Shailendra et al.[21]	Extractive	Multi-document text summarization	FIRE 2011 dataset	BPSO with improved genetic algorithm	Performance of BPSO was higher than other discussed methods
Sunil et al.[16]	Extractive	Document Summarization	FIRE Dataset	Statistical sentence scoring method	Performance of proposed method was higher than other methods
Sumanlata et al. [22]	Extractive	Document Summarization	Dataset from Kaggle	Political Elephant Herding Optimization	Rouge Score- 0.94
Bhansali et al.[23]	Abstractive	Text Summarization	Hindi News Kaggle-Dataset	Transformer	R-1 Score- 0.59
Shah et al. [20]	Abstractive	Text Summarization	Own Dataset	LSTM with Attention	Rouge Score- 0.675
Dhaval et al.[24]	Abstractive	Text Summarization	Own Dataset	Fine-tuning of Pre-trained models	Rouge Score-0.61
Arjit et al[25]	Abstractive and Fine-tuning	Document Summarization	ILSUM	Fine-tuning of Pre-trained models	Rouge Score- 0.54

3. Methodology

The challenge of obstructively summarizing Hindi language text is difficult. In the proposed work, we have scrapped two large datasets from reputed news websites and used one publicly available dataset. After preprocessing of the data, we have fine-tuned four pre-trained models i.e. Someman/BART-hindi, facebook/mbart-large-50, indicBART and mT5 models. For measuring the performance, ROUGE score, BLEU score, BERT score and semantic similarity scores have been used. Following figure 1 shows the proposed workflow of Hindi text summarization using fine-tuning.

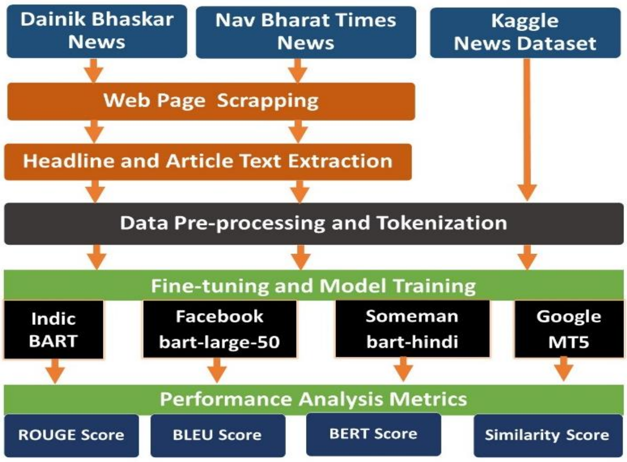

Fig. 1. Proposed workflow of Hindi text summarization using fine-tuning

3.1. Dataset

For our research work, we have collected two datasets, namely NTS_News and DB_News, which were acquired from the news websites Navbharat Times and Dainik Bhaskar. The databases consist of news stories and headlines encompassing various subjects, such as national news, state news, crime reports, sports, politics, and Bollywood, among others. In addition, we have included a dataset[26] that is accessible to the public. This dataset contains Indian news stories and their accompanying titles, which have been obtained from various domains such as national news, crime, entertainment, sports, and others. This dataset includes articles and their headlines collected from Hindi news websites. Following Table 2 and Table 3 shows statistics of dataset used.

Table 2. Dataset Description

Dataset	*Number of article and headline pairs*	
	Train Data	*Test Data*
NTS_News	3,39,048	37,672
DB_News	3,10,154	34,463
HINDI_TEXT_SUMM[26]	1,43,867	35,968

Table 3. Dataset statistics

Lengt*	*NTS_News*				*DB_News*				*HINDI_TEXT_SUMM[26]*			
	Train Data		*Test Data*		*Train Data*		*Test Data*		*Train Data*		*Test Data*	
	A*	H*	A*	H*	A*	H*	A*	H*	A*	H*	A*	H*
Max.	4731	183	4168	45	4914	30	3420	28	16335	226	10080	24
Min.	11	6	0	6	17	1	18	3	1	6	2	6
Mean	327.4	14.2	313.7	14.1	278.5	13.0	276.6	13.1	363.2	11.6	362.7	11.6

In number of words, A- Article, H*- Headline

3.2. Data Pre-processing

Data preprocessing is a crucial process that is necessary to prepare data for analysis, hence improving their efficiency and accuracy. In order to streamline this pre-processing task, we utilized two tools, specifically Genism and iNLTK[27]. During the early stages of data pre-processing, we started by removing HTML tags from the text. Consequently, all line breaks and additional spaces were rigorously eliminated. Furthermore, all emojis were deliberately omitted from the dataset to guarantee the integrity and appropriateness of the data for our machine learning models.

3.3. Models

In this research work, we have fine-tuned four pre-trained models, namely Someman/bart-hindi, facebook/mbart-large-50, indicBART, and mT5, on three different datasets. These datasets include two our collected datasets NTS_News and DB_News and one publically available datasets Hindi_Text_SUMM downloaded from kaggle. The major goal of this fine-tuning procedure is to provide these language models i.e. Someman/BART-hindi, facebook/mbart-large-50, indicBART and mT5, the capacity to produce content of exceptional quality, closely matching with the distinctive linguistic traits, contextual subtlety, and stylistic aspects intrinsic to the defined domains of application.

3.3.1. Someman/bart-hindi

Someman/bart-hindi is a fine-tuned version of facebook/bart-base on the Someman/hindi-summarization dataset. It is a large language model (LLM) that can be used for a variety of tasks, including text summarization, machine translation, and question answering. It is a powerful tool for processing and generating human language, and it has been used to achieve state-of-the-art results on a number of natural language processing benchmarks. Someman/bart-hindi is a BART model, which stands for Bidirectional and Auto-Regressive Transformers. BART models are a type of neural network that is trained on a massive dataset of text and code. This training allows the model to learn the statistical relationships between words and phrases, which it can then use to generate text, translate languages, and answer questions. Someman/bart-hindi is specifically trained on the Someman/hindi-summarization dataset, which is a collection of Hindi text and its corresponding summaries. This training allows the model to learn how to summarize Hindi text in a comprehensive and informative way.

3.3.2. facebook/mbart-large-50

The mBART-50 was trained to recognize many languages using Multilingual Denoising Pre-training. The Sequence-to-Sequence model mBART-50 supports several languages. It demonstrates, how multilingual fine-tuning may be used to generate translation models for many languages. Pre-trained models are tweaked simultaneously in several directions rather than just one. To accommodate multilingual machine translation models of 50 languages, the original mBART model was expanded to incorporate an additional 25 languages, creating mBART-50. The model integrates N languages by concatenating the data as follows: D = D1,..., DN where each Di represents a collection of monolingual documents in language i. Two methods are used to noisy the source documents: the first randomly shuffles the sequence of the original phrases, and the second is a new in-filling method that replaces long stretches of text with a single mask token. The original text is then recreated using the model. By randomly selecting a span length from a Poisson distribution (= 3.5), 35% of each instance's words are hidden. The original text with a single place offset is the decoder input.

3.3.3. IndicBART

IndicBART[28] is a transformer model that has been pre-trained on a large dataset of text in 11 Indic languages and English, making it capable of understanding and generating text in many languages. Google AI created and launched it in 2022. IndicBART is a robust model suitable for many natural language processing (NLP) applications, such as machine translation, text summarization, question answering, and text synthesis. An important benefit of

IndicBART is its specialized training in Indic languages. This implies that it possesses an enhanced ability to comprehend the distinct characteristics of Indic language usage, including the intricate morphology and syntax of these languages. Consequently, IndicBART outperforms other pre-trained models that lack specific training on Indic languages, leading to superior performance on NLP tasks for these languages.

3.3.4. multilingual Text-to-Text transfer transformers (mT5)

mT5[29], also known as Multilingual T5, is an expanded version of the T5 (Text-to-Text Transfer Transformer) model created by Google AI. This robust natural language processing (NLP) model is specifically built to process text data in several languages, making it an invaluable tool for multilingual NLP jobs. mT5 is constructed using the transformer architecture and adopts the text-to-text framework, where it considers different NLP jobs as challenges of generating text. The model demonstrates exceptional performance in both the pre-training and fine-tuning phases. During pre-training, it is trained on a vast multilingual text dataset to develop a robust foundation of language comprehension. Subsequently, it undergoes fine-tuning to specialize in certain natural language processing tasks. The exceptional performance of mT5 in settings involving few-shot and zero-shot learning renders it remarkably versatile in tackling new tasks with limited examples or context. The multilingual features of the system are advantageous for cross-lingual applications, enabling developers and researchers to efficiently design NLP models that are effective in several languages. mT5 is well-suited for a range of NLP tasks, such as machine translation, document categorization, sentiment analysis, text summarization, and question-answering. This makes it a versatile solution for processing languages in multiple languages. mT5 utilizes the Transformer architecture, which comprises of an encoder and a decoder as its main components. The encoder operates on a series of input tokens, generating a series of hidden states, while the decoder utilizes the encoded hidden states to produce a series of output tokens, adhering to the text-to-text paradigm. The mT5 model has a remarkable 11 billion parameters, which enhance its ability to comprehend and produce text in various languages. mT5 possesses a vast and comprehensive lexical repository of 101,292 tokens, enabling it to effectively process text in multiple languages. The competency of mT5 is based on its thorough pre-training on a vast dataset consisting of 1 trillion pieces of text from 101 languages. mT5 is equipped with a strong foundation in comprehending and generating languages because to its extensive exposure to many linguistic contexts.

4. Results and Discussion

All three models have generated the summarization of the Hindi text. We have used four evaluation matrices i.e. ROUGE score[30], BLEU score[31], BERT Score[32] and Sematic Similarity[33] score.

4.1. ROUGE Score

The ROUGE score is a quantitative metric ranging from 0 to 1 that evaluates the degree of similarity between the candidate text and reference texts. A lower ROUGE score, approaching 0, signifies

a diminished likeness between the candidate and references, indicating a lack of similarity. On the other hand, a higher ROUGE score, close to 1, indicates a strong resemblance between the candidate text and the reference texts, indicating a stronger match. ROUGE scores are calculated by assessing the presence of unigram, bigram, trigram, and the identification of the longest common sequence between the texts.

4.2. BLEU Score

The BLEU score provides a numerical value ranging from 0 to 1, which indicates the similarity between the candidate text and the reference texts. When the BLEU score approaches 0, it indicates that there is little similarity between the candidate and references, indicating a low level of resemblance. Conversely, a BLEU score approaching 1 indicates a significant similarity between the candidate text and the references, indicating a high level of agreement. The BLEU score is commonly denoted as BLEU-N, with "N" indicating the exact n-grams taken into account, usually ranging from 1 to 4 grams.

4.3. BERT Score

The BERT Score statistic quantifies the degree of similarity between machine-generated text and a reference text. The system employs a pre-trained BERT model for the purpose of calculating similarity, where higher scores are indicative of superior quality in the output text.

4.4. Semantic Similarity Score

SSEM is a Python module that provides evaluation metrics for NLP text generation tasks in multiple languages. The main focus of the library is to ascertain the semantic similarity between the generated content and the reference information. The program supports multiple distance metrics, such as cosine similarity, Euclidean distances, and Pearson correlation. The library is built upon the widely-used Hugging Face Transformers library and is compatible with any pre-trained transformer model. In addition, it enables parallel processing to enhance computational speed and offers many evaluation levels, such as sentence-level, token-level, and similarity based on Latent Semantic Indexing (LSI).

The differences in scores obtained after fine-tuning several models are displayed in Table 4, using the evaluation criteria indicated earlier.

Table 4. ROUGE, BLEU, BERT and Similarity score of generated summaries on different dataset after fine-tuning the model.

Model	R-1		R-2		R-L		BLEU Score	BERT Score	Semantic Similarity Score
	F1-score	Recall	F1-score	Recall	F1-score	Recall			
NTS_News Dataset									
Someman/bart-hindi	0.24	0.22	0.08	0.07	0.21	0.20	0.06	0.74	0.79
facebook/mbart-large-50	0.43	0.44	0.24	0.25	0.39	0.40	0.19	0.79	0.79
indicBART	0.23	0.30	0.09	0.13	0.19	0.26	0.05	0.69	0.78
mT5	0.25	0.27	0.09	0.10	0.21	0.24	0.06	0.71	0.79
DB_News Dataset									
Someman/bart-hindi	0.20	0.18	0.06	0.05	0.18	0.16	0.04	0.72	0.80
facebook/mbart-large-50	0.33	0.33	0.16	0.16	0.30	0.30	0.12	0.77	0.80
indicBART	0.22	0.31	0.08	0.12	0.19	0.26	0.05	0.70	0.80
mT5	0.21	0.24	0.06	0.07	0.18	0.21	0.05	0.70	0.79
HINDI_TEXT_SUMM[26]									
Someman/bart-hindi	0.22	0.20	0.06	0.06	0.19	0.18	0.05	0.73	0.78
facebook/mbart-large-50	0.39	0.39	0.19	0.19	0.34	0.34	0.14	0.78	0.79
indicBART	0.24	0.36	0.08	0.14	0.20	0.31	0.05	0.70	0.79
mT5	0.25	0.29	0.09	0.11	0.21	0.24	0.06	0.72	0.79

From table 4, we can see that performance of Someman/bart-hindi model and indicBART and mT5 model is consistently low in terms of R1 score, R2 score, RL score and BLEU Score. However high similarity score of these model suggest that they can generate the text which is semantically similar. Among all the four models under study, face/mbart-large-50 model performs better than other models in terms of R1 Score, R2 Score, RL score, BLEU score and semantic similarity Score. The strong Semantic Similarity Scores indicate that the produced text from all models retains semantic similarity with the reference even when n-gram overlap is minimal, which may be crucial for some applications. Following table 5 shows the sample of gold summary and summary predicted using all four pre-trained models after fine-tuning. From table 5, it also very clear that facebook/mbart-large50 model generated summaries are correct and can be accepted. Summaries generated by indicBART and T5 model are incomplete. Summaries generated by Someman/bart-hindi are not fully correct. So, finally it can be concluded that facebook/mbart-large-50 model performs better than other models in all respect for Hindi text summarization.

Table 5. Gold summary and Generate summary after fine-tuning the model

Gold Summary	Someman/ bart-hindi	facebook/mbart-large-50	indicBART	mT5
झारखंड विधासभा चुनाव: बृहस्पतिवार को 17 सीटों पर मतदान	झारखंड विधानसभा चुनाव :17 सीटों पर मतदाता बृहस्पत करेगा	झारखंड विधानसभा चुनाव के तीसरे चरण में 17 सीटों पर मतदान	राज्य रांची, 11 दिसंबर :भाषाः झारखंड विधानसभा चुनाव के तीसरे चरण में बृहस्पतिवार को 17 सीटों पर मतदान होगा। चुनाव निर्वाचन निर्वाचन आयोग ने	झारखंड विधानसभा चुनाव के तीसरे चरण में बृहस्पतिवार को 17 सीटों पर मतदान होगा
राजनाथ ने डीआर डीओ और घरेलू रक्षा उद्योगों के बीच सामंजस्य बनाने की अपील की	डीआरडिओ, घरेलू रक्षा उद्‍योग आत् मनि के साथ तलाशना चाहता है: राजसभा	डीआरडीओ, रक्षा उद्योगों को आत्मनिर्भरता का लक्ष्य हासिल करने के लिए साथ मिल कर नए तरीके तलाशने चाहिए : राजनाथ	स्वातंत्र्र नयी दिल्ली, 22 नवंबर (भाषा) रक्षा मंत्री राजनाथ सिंह ने शुक्रवार को कहा कि रक्षा अनुसंधान एवं विकास संगठन (डीआरडीओ)	रक्षा मंत्री राजनाथ सिंह ने कहा है कि देश में आत्मनिर्भरता के लिए नए तरीके तलाशने

5. Conclusion

The field of news headline generation, which falls under the umbrella of automatic text summarization, serves a vital function in the process of reducing lengthy texts into brief summaries. This process eventually benefits readers by saving them time. The work of text summarizing has been mostly centered around the English language, resulting in a lack of available summary tools for other languages. This study aimed to fill the existing research gap by incorporating datasets sourced from important Hindi news outlets, namely Dainik Bhaskar and NavBharat Times, in addition to a publically accessible dataset. In this study, we conducted fine-tuning on four pre-trained language models, namely Someman/bart-hindi, facebook/mbart-large-50, indicBART, and mT5. Subsequently, we performed a comprehensive evaluation of their performance using various metrics such as ROUGE score, BLEU score, BERT score, and semantic similarity score. The observed consistent superiority of the facebook/mbart-large-50 model across various measures underscores its potential to substantially enhance automated summarization systems, hence enhancing the accessibility and efficiency of information retrieval and understanding for the Hindi-speaking community. The future prospects of abstractive Hindi text summarization encompass the enhancement of natural language generation methodologies, the refinement of strategies for handling intricate grammar and semantics, and the tailoring of summaries to cater to a wide range of user requirements and domains. Additionally, attention will be given to addressing linguistic and cultural subtleties that are unique to the Hindi language.

Acknowledgements

None.

Conflicts of interest

None.

References

[1] A. P. Widyassari *et al.*, "Review of automatic text summarization techniques & methods," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1029–1046, Apr. 2022, doi: 10.1016/J.JKSUCI.2020.05.006.

[2] G. Sharma and D. Sharma, "Automatic Text Summarization Methods: A Comprehensive Review," *SN Comput. Sci.*, vol. 4, no. 1, pp. 1–18, Jan. 2023, doi: 10.1007/S42979-022-01446-W/METRICS.

[3] Y. Kumar, K. Kaur, and S. Kaur, "Study of automatic text summarization approaches in different languages," *Artif. Intell. Rev.*, vol. 54, no. 8, pp. 5897–5929, Dec. 2021, doi: 10.1007/S10462-021-09964-4/METRICS.

[4] A. V. Pradeepika Verma, "Accountability of NLP Tools in Text Summarization for Indian Languages," *J. Sci. Res.*, vol. 64, no. 1, pp. 358–363, 2020.

[5] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Syst. Appl.*, vol. 165, p. 113679, Mar. 2021, doi: 10.1016/J.ESWA.2020.113679.

[6] A. K. Yadav *et al.*, “Extractive text summarization using deep learning approach,” *Int. J. Inf. Technol.*, vol. 14, no. 5, pp. 2407–2415, Aug. 2022, doi: 10.1007/S41870-022-00863-7/METRICS.

[7] E. R. Mahalleh and F. S. Gharehchopogh, “An automatic text summarization based on valuable sentences selection,” *Int. J. Inf. Technol.*, vol. 14, no. 6, pp. 2963–2969, Oct. 2022, doi: 10.1007/S41870-022-01049-X/METRICS.

[8] S. Mandal, G. K. Singh, and A. Pal, “Single document text summarization technique using optimal combination of cuckoo search algorithm, sentence scoring and sentiment score,” *Int. J. Inf. Technol.*, vol. 13, no. 5, pp. 1805–1813, Oct. 2021, doi: 10.1007/S41870-021-00739-2/METRICS.

[9] P. J. Goutom, N. Baruah, and P. Sonowal, “An abstractive text summarization using deep learning in Assamese,” *Int. J. Inf. Technol.*, vol. 15, no. 5, pp. 2365–2372, Jun. 2023, doi: 10.1007/S41870-023-01279-7/METRICS.

[10] G. B. Mohan and R. P. Kumar, “Lattice abstraction-based content summarization using baseline abstractive lexical chaining progress,” *Int. J. Inf. Technol.*, vol. 15, no. 1, pp. 369–378, Jan. 2023, doi: 10.1007/S41870-022-01080-Y/METRICS.

[11] K. Rudra, S. Banerjee, N. Ganguly, P. Goyal, M. Imran, and P. Mitra, “Summarizing situational tweets in crisis scenario,” *HT 2016 - Proc. 27th ACM Conf. Hypertext Soc. Media*, pp. 137–147, Jul. 2016, doi: 10.1145/2914586.2914600.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3104–3112, Sep. 2014, doi: 10.48550/arxiv.1409.3215.

[13] V. Dalal and L. Malik, “Semantic graph based automatic text summarization for hindi documents using particle swarm optimization,” *Smart Innov. Syst. Technol.*, vol. 84, pp. 284–289, 2018, doi: 10.1007/978-3-319-63645-0_31/COVER.

[14] M. L. Joshi, N. Joshi, and N. Mittal, “SGATS: Semantic Graph-based Automatic Text Summarization from Hindi Text Documents,” *Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 6, Sep. 2021, doi: 10.1145/3464381.

[15] A. Jain, A. Arora, J. Morato, D. Yadav, and K. V. Kumar, “Automatic Text Summarization for Hindi Using Real Coded Genetic Algorithm,” *Appl. Sci.*, vol. 12, no. 13, 2022, doi: 10.3390/app12136584.

[16] S. Dhankhar and M. K. Gupta, “A statistically based sentence scoring method using mathematical combination for extractive Hindi text summarization,” *J. Interdiscip. Math.*, vol. 25, no. 3, pp. 773–790, 2022, doi: 10.1080/09720502.2021.2015096.

[17] S. S. Aote, A. Pimpalshende, A. Potnurwar, and S. Lohi, “Binary Particle Swarm Optimization with an improved genetic algorithm to solve multi-document text summarization problem of Hindi documents,” *Eng. Appl. Artif. Intell.*, vol. 117, p. 105575, 2023, doi: https://doi.org/10.1016/j.engappai.2022.105575.

[18] R. Bhargava, G. Sharma, and Y. Sharma, “Deep Text Summarization using Generative Adversarial Networks in Indian Languages,” *Procedia Comput. Sci.*, vol. 167, pp. 147–153, Jan. 2020, doi: 10.1016/J.PROCS.2020.03.192.

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

[20] A. Shah, D. Zanzmera, and K. Mehta, “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.

[21] S. S. Aote, A. Pimpalshende, A. Potnurwar, and S. Lohi, “Binary Particle Swarm Optimization with an improved genetic algorithm to solve multi-document text summarization problem of Hindi documents,” *Eng. Appl. Artif. Intell.*, vol. 117, p. 105575, Jan. 2023, doi: 10.1016/J.ENGAPPAI.2022.105575.

[22] S. Bandari and V. V. Bulusu, “Feature extraction based deep long short term memory for Hindi document summarization using political elephant herding optimization,” *Int. J. Intell. Robot. Appl.*, vol. 7, no. 1, pp. 103–118, 2023, doi: 10.1007/s41315-022-00237-z.

[23] R. Bhansali, A. Bhave, G. Bharat, V. Mahajan, and M. L. Dhore, “Abstractive Text Summarization of Hindi Corpus Using Transformer Encoder-Decoder Model,” *Smart Innov. Syst. Technol.*, vol. 333, pp. 171–185, 2023, doi: 10.1007/978-981-19-8094-7_13/COVER.

[24] D. Taunk and V. Varma, “Summarizing Indian Languages using Multilingual Transformers based Models,” *CEUR Workshop Proc.*, vol. 3395, pp. 435–442, Mar. 2023, Accessed: Sep. 29, 2023. [Online]. Available: https://arxiv.org/abs/2303.16657v1

[25] and S. S. Agarwal, Arjit, Soham Naik, “Abstractive Text Summarization for Hindi Language using IndicBART,” *Work. Notes FIRE 2022-Forum Inf. Retr. Eval.*, 2022.

[26] “Hindi Text Short Summarization Corpus | Kaggle.” https://www.kaggle.com/datasets/disisbig/hindi-text-short-summarization-corpus (accessed Feb. 02, 2023).

[27] G. Arora, “iNLTK: Natural Language Toolkit for Indic Languages,” pp. 66–71, Sep. 2020, doi: 10.18653/v1/2020.nlposs-1.10.

[28] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. M. Khapra, and P. Kumar, “IndicBART: A Pre-trained Model for Indic Natural Language Generation,” *Proc. Annu. Meet. Assoc. Comput. Linguist.*, vol. 2, pp. 1849–1863, 2022, doi:

10.18653/V1/2022.FINDINGS-ACL.145.

[29] L. Xue *et al.*, "mT5: A massively multilingual pre-trained text-to-text transformer," *NAACL-HLT 2021 - 2021 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Conf.*, pp. 483–498, Oct. 2020, doi: 10.18653/v1/2021.naacl-main.41.

[30] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[31] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," *Proc. 40th Annu. Meet. Assoc. Comput. Linguist. - ACL '02*, pp. 311–318, 2002, doi: 10.3115/1073083.1073135.

[32] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv Prepr. arXiv1904.09675*, 2019.

[33] "GitHub - TechyNilesh/SSEM: SSEM is a semantic similarity-based evaluation library for natural language processing (NLP) text generation tasks. It supports various similarity metrics and evaluation levels, and is compatible with any Hugging Face pre-trained transformer model." https://github.com/TechyNilesh/SSEM (accessed Sep. 25, 2023).