International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(3s) 2026 pp. 76-87
https://doi.org/10.70917/ijcisim-2026-2280
Article

Enhancing Abstractive Text Summarization for Indic Scripts using Transformer-Based Models

Ved Kumar Gupta[1, *], Dr. Hare Ram Sah[2]

[1, *]Computer Science & Engineering Department, Institute of Engineering & Technology, SAGE University, Indore, India

[2]Computer Science & Engineering Department, Institute of Engineering & Technology, SAGE University, Indore, India

* Correspondence author: vkgupta.491@gmail.com

Abstract: There has been a rapid growth of digital textual content which has increased the importance of automatic summarization to extract useful information. So far, while it has seen considerable improvement in the case of English and other high-resource languages, summarization in Indian languages still falls far behind compared to these languages due to the diversity, morphological richness and lack of ample annotated dataset. In this work, we have tested how effectively multilingual transformers can perform abstractive summarization for Hindi, Gujarati and Marathi languages.

The fine-tuned versions of IndicBART and mBART are tested to produce brief summaries with the retention of meanings of the documents. For Hindi and Gujarati, experiments were done with the ILSUM 2.0 benchmark dataset while for Marathi experiments the XL-Sum Marathi corpus was used. The obtained models were evaluated with the evaluation metrics such as Rouge-1, Rouge-2, Rouge-L, BLEU and BERTScore, and compared with the baseline models based on Seq2Seq-LSTM that have widely been used for the summarization tasks on Indic languages. Experimental analysis results show significant improvements across all the evaluated measures. For Hindi language, indicBART gives maximum performance (ROUGE-L, ROUGE-1 and ROUGE-2 values of 0.6176, 0.6449, 0.4763 respectively).

For the Gujarati language, mBART achieves maximum performance with Rouge-1, Rouge-2 and Rouge-L scores 0.6439, 0.4069 and 0.5899 respectively.

Experiments on Marathi provide proof that transformer models can be used even in the low-resource situations. In this case, indicBART achieved optimum results. These results prove that language-aware fine-tuning enhance more lexical overlapping, semantic meaning and context conservation capabilities of models.

Keywords: Abstractive Text Summarization, IndicBART, mBART, Indian Languages, Transformer Models, Hindi, Gujarati, Marathi.

1. Introduction

The unprecedented growth in online news, digital archives, social networks and learning repositories leads to an enormous amount of unstructured textual data. As a result, for a user it becomes increasingly challenging to retrieve meaningful information from such collections of documents. To tackle these difficulties, automatic text summarization tries to generate short versions of given documents that retain important information and overall semantics of the source.

Automatic summarization systems reduce the reading effort and facilitate swift access to the information in several domains like education, journalism, governance, health care and digital libraries. The techniques used for text summarization are primarily divided into two categories i.e. Extractive and abstractive. Extracting of important sentences or phrases directly from source documents results in extractive systems. While the extractive methods can produce factually relevant summaries they can also suffer from low coherency and some form of redundancy. An abstractive approach, in contrast, generates sentences which describe the central meaning of original documents and thereby produce abstract kind of summarization. The abstractive approach is more challenging to achieve a deeper understanding about contexts, semantics, and linguistic structure of the given text.

Over the past several years, Deep Learning technologies have revolutionized the field of abstractive text summarization. For example, the neural Seq2Seq models that exploit RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory) and Attention mechanism show great improvement over the statistical models; however these architectures face difficulty with long sequences and cannot model distant dependencies efficiently. Transformer-based architectures have shown success to capture global long range relationships between different parts of input sequence through self-attention mechanism.

There is a relative limited research progress on the summarization task of Indian languages in comparison to English and other high-resource languages. Challenges like morphologically rich nature, flexible sentence formation, characteristics specific to scripts, limited availability of benchmark datasets etc make the task more

complicated for languages such as Hindi, Gujarati and Marathi. In recent past, multilingual language models with the state-of-art neural architectures offered exciting new directions in the area of Indic language processing. Models such as IndicBART and mBART make use of multilingual pretraining strategy and transfer learning techniques to perform across several Indic languages. The availability of benchmark datasets like ILSUM and XL-Sum provided the way to boost research progress in Indic summarization which allow us to train/finetune state-of-art neural architectures on high quality data. With these large scaled datasets, we can utilize pre-trained multilingual models and further train them for task-specific summarization.

The research work undertaken investigates the effectiveness of IndicBART and mBART models to perform task of abstractive summarization in three major Indian languages viz. Hindi, Gujarati, and Marathi. The evaluation for Hindi and Gujarati languages was carried out on the ILSUM 2.0 corpus, whereas for Marathi, XL-Sum corpus has been used. The transformer-based models have been compared against the traditional Seq2Seq-LSTM architecture with different performance metrics: ROUGE, BLEU and BERTScore. Extensive experiments carried out to analyze the impact of multilingual pretraining and language-specific finetuning on summary generation of Indian languages.

Here are the major contributions of this study:

1. Design and implementation of transformer-based abstractive summarization for three Indian languages: Hindi, Gujarati, and Marathi.
2. Fine-tuning of IndicBART and mBART models for abstractive summarization using pre-defined datasets for the concerned Indian languages.
3. Comparison of developed transformer-based summarization models against a Seq2Seq-LSTMbaseline with diverse evaluation metrics (e.g., ROUGE, BLEU, BERTScore).
4. Analysis of the cross-lingual knowledge transfer capability of multilingual transformer models in low-resource language scenario.
5. Performance evaluation based on lexical as well as semantic features of summary for enhanced summary assessment.

2. Related Work

Automatic text summarization has become one of the leading research areas in Natural Language Processing, due to the explosion of digital information across news media, healthcare, governance, education, and online social networking[1],[35]. The primary objective of text summarization system is to provide an adequate and succinct version of the original document while containing the information about essential entities, facts and semantic meaning[1],[15]. Currently, it is divided into two types: extractive summary, which selecting significant parts of the document to retain the original information, and abstractive summary, which generate novel sentences summarizing the source[1],[24],[25].

In the past, text summarization has benefited from statistical and heuristic-based approaches that primarily consisted of extracting important sentences based on sentence location, term frequencies, keyword presence, and graph-based sentence rankings[1],[15].

These methods usually deliver reasonable quality summarization for formal documents, but they have suffered from lack of semantic integrity in their outputs. In addition, they lack good adaptability to many formal languages like Indian languages that aremorphologically rich and contain free word order[26],[27]. The introduction of neural network approaches such as thesequence-to-sequence (seq2seq) modelarchitecture [11] enabled the transformation between natural language text in a flexible and scalable way. Further improvements based on Long Short-Term Memory (LSTM) networks and attention mechanisms have greatly enhanced the performance and generation of cohesive summaries[12],[24],[25].

However, due to the serial nature of recurrencies, the sequence to sequence model fails to handle long distance dependency information and generate accurate long coherent texts[2],[16].

Transformer architecture[2] successfully resolved the long dependency issues by converting recurrent computation to self-attention mechanism, which simultaneously calculated relation among the whole document. In recent years, the attention-based neural network is utilized for machine translation[3],[6],[34],text generation[3],[6], question answering[3],[6], and summarization[3],[6],[34]. The Transformer-based encoder-decoder models such as BART[7],T5[30] and mT5[30] have proved their state-of-the-art summarization capabilities owing to their scaleable pretraining strategies.

The multilingual encoder-decoder models that utilize large-scale multilingual corpus have opened opportunities for the application of abstractive summarization to the low-resource languages. MBART is one of

the such popular model which employ a sequence-to-sequence architecture trained over the parallel corpus in multiple languages. The multilingual approach facilitates cross-lingual knowledge transfer by sharing the representation among different languages. Studies[4],[32] reported competitive performances on the summarization task by using mBART model.

Many research studies have explored and adapted transformer models for Indian language tasks. The challenges specific to Indian languages include rich and complex morphology, diverse scripting systems, and scarcity of labeled corpora[8],[9]. In an attempt to cope with the inherent complexities of Indic languages, several studies focused on adapting the Indic languages specific transformer model IndicBART for text generation tasks such as summarization [5],[10],[31]. The IndicBART[5]model is trained on massive multilingual data and specially developed to provide the language specific characteristics. Experiments[5],[31] demonstrated promising performance ofIndicBART for multiple Indian languages with excellent adaptation to local features than standard multilingual models.

Development of standard datasets such as ILSUM [13],[14] have boosted further research in Indic languages summarization. ILSUM initiative comprises of a group of Hindi, Gujarati, Marathi and other Indian languages datasets prepared with the intention of standardizing the evaluation process across different studies. Recent experimental results have suggested that utilizing pretrained multilingual models and further finetuning on these datasets improved the summarization performances Significantly[14],[26],[32]. The evaluation metrics used to measure summarization quality have also undergone several developments.

Initially, recall-oriented understudy for gisting evaluation (ROUGE) metrics have been widely used to evaluate summarizing works based on lexical overlap with the reference summaries[15]. The Bilingual Evaluation Understudy (BLEU) was initially developed for evaluating the machine translation outputs but has been widely applied to text summarization research[16],[18]. Recently, attention is given tosemantic evaluationbased metrics like BERTScore[17],[29] since they measure the similarity between model predicted and human generated abstract using transformer language models. Lexical and semantic evaluation metrics jointly provides reliable evaluation for summarized outputs, and more effective for morphologically rich languages[19],[27].

Several aspects of Indic language summarization are yet under explored such as focused research on very few languages, small scale evaluation datasets orsingle model validation. Lack of detailed comparison on multilingual and Indic-specific transformer models in a common benchmark framework remains a void[9],[26]. Moreover,Marathi summary research is much lesser compared to its highly spoken counterpart,Hindi despite more research focus now on the availability of benchmark datasets. Inspired by these open research questions, the present study investigates and compares IndicBART and mT5 models in abstractive summarization of three prominent Indic languages - Hindi, Gujarati and Marathi.

The Hindi and Gujarati datasets utilized for summarization task belong to the ILSUM 2.0 corpus [14], while evaluation of Marathi Summarization is performed using the XL-Sum[20]. Comprehensive experiments have been conducted comparing their performance on both lexical and semantic metrics i.e. ROUGE, BLEU and BERTScore for establishing their suitability in low-resource and multilingual summarization.

3. Research Methodology

In this study, a multilingual abstractive text summarization system for Hindi, Gujarati and Marathi languages has been developed using transformer based encoder-decoder networks. IndicBART [5] and mBART [4] have been used as primary models due to its suitability in multilingual text generation and sequence to sequence tasks. The framework integrates language-specific preprocessing steps, transformer models fine-tuning, summary generation and quantitative evaluation for producing linguistically coherentSummaries in low resource Indian Languages.

3.1 Proposed Framework

The whole framework begins with collecting document summary pairs from various datasets. Hindi and Gujarati documents have been obtained from ILSUM 2.0 corpus, whereas the Marathi documents have been taken from theXL-Sum[13, 14] dataset. Raw documents are preprocessed and given as input to the transformer model in the form of text normalisation, script-specific tokenisation, and sub-word segmentation [22, 28]. The tokenised documents are then fed to IndicBART[5] and mBART[4] architectures for supervised fine-tuning. During the training process the models are trained to map source document to target human written summaries by minimizing generation error between the output predictions and reference sequence[3, 4]. After, which the fine-tuned models produces abstractive summaries through auto regressive decoding and are evaluated on the basis of ROUGE[15], BLEU[16] and BERTScore[17]. The summary generation work flow consists of the following steps: 1. Dataset Preparation. 2. Text Normalization and Tokenization. 3. Training of models. 4. Summary generation. 5. Evaluation.

3.2 Model Architecture

Sequence to sequence problem for abstractive summarization. Our proposed approach also follows the encoder-decoder transformer architecture developed by Vaswaniet al[2] for sequence transduction solutions. Transformer based approach, based on self attention mechanism[2, 3]. It takes sequence of tokens as a s Inputd ocument. N represent the number of tokens in the input sequence $X = \{x_1 , x_2,..., x_n\}$. Our goal is to train a model capable of translating X into the sequence of tokens in the summary, $Y=\{y_1 , y_2 , ..., y_m\}$. We try to maximize the probability of the target sequence Y conditioned on the source sequence X as:

$$P(Y|X) = \prod P(y_t \mid y_1,...,y_{t-1},X)$$

Here, y_i denotes the token atthesi tion in the output summary. The encoder decodes the representation of the input sequence. Decoder decodes summary tokens one by one by attending over encoded sequences and previous summary tokens[3, 4]. The encoder of a transformer architecture comprises of the following layers. Decoder comprises of following layers.

3.3 IndicBART

IndicBART[5] is multilingual text generation model for Indian languages which extends BART architecture. IndicBART have been extensively pre trained on a multilingual indic text dataset consisting of numerous languages and a diverse set of scripts and multiple Indian language family. The encoder part in IndicBART consists of same stacked self attention layers as the encoder in transformers and these self-attntion layers are stacked repeatedly. Feed-forward neural networks layer consists of fully connected layers with RELU as activation and Layer Normalization is also applied. Decoder part generate summary tokens individually by attention mechanism over encoded sequences and previously generated tokens[3].

Features of INDICBART is as follow:

• Better adaptation to Indic linguistic patterns[5].

• Improved handling of morpho -syntactic variations and derivational richness[8, 10].

• Broad vocabulary coverage on Indian Scripts.

• Less semantic Drift in text generated[31].

3.4 mBART

mBART[4] is a denoising sequence-to-sequence model that is trained across and beyond hundreds of languages, including Indian languages like Hindi, Gujarati, etc. MBART[4] use same transformer based encoder-decder architecture as that used in Bart but is pre-trained with shared vocabulary across language through large multilingual corpora. Encoder-decoding mechanism captures cross-lingual transfer ability which allows it to generate a text even with limited amounts of training data in specific languages like Indian languages[20, 21].

Following are the advantages of mBART that help potential for this task:

• Excellent cross-lingual transferable knowledge[4].

• Powerful representation of contexts[20].

• Handle low resource setup perfectly.

• Best in terms of quality sequence generation[4].

3.5 Training Strategy

The proposed models IndicBART and mBART were finetuned using supervised learning on Benchmark Datasets [13, 14]. Input and output sequence, in the fine-tuning process are consisted of document and reference summary, respectively. Teacher forcing method isused for stabilizing the training and speed up convergence[11]. In this method during decoding, actual target word isused as input to predict next word. This reduce the chance of accumulation of error at each decoding step of prediction [12]. We optimize cross-entropy loss function by training models. The loss is calculated as

$$L = - (1/N) \Sigma \log P(y_t|X,y<t)$$

Where N denotes total number of tokens in the target and $P(y_t|X,y<t)$ is the probability assigned to the correct target token at stept[3,11].

Training is performed by Adam optimiser, which is a low-memory and effective Optimizer that combines ideas of adaptive learning rates and RMS Prop [33]. We are using early stopping criteria based on the model's performance on the validation set, to avoid over-fitting [19]. While generating summaries during Inference mode, we are using the BeamSearch decoding method for generating summaries that generate high-quality summaries [23]. In contrast to greedy decoding approach beam search selects not only most likely word but the entire summary sentence by selecting best summary among all possibilities using summed probability scores. The training process allows the IndicBART and mBART to leverage their pretrained cross-lingual intelligence with the linguistic constraints and domain specificity of monolingual Hindi, Gujarati, and Marathi summarization [5], [10], [31], [32].

4. Experimental Setup

In this section, we detail the datasets and experimental setup for the evaluation of IndicBART and mBART for multilingual abstractive summarization. These include the training data, pre-processing techniques used, the baseline system, training configurations, and the evaluation metrics. We aimed to provide an equal opportunity of evaluation for the Hindi, Gujarati, and Marathi languages. We kept the evaluation methodology the same for all the Transformer-based models to compare outcomes with the best of knowledge.

4.1 Dataset Description

We have conducted our experimental analysis using two benchmark datasets built for the purpose of abstractive summarization in Indian languages. The ILSUM 2.0 dataset has been utilized for Hindi and Gujarati experiments while the XL-Sum Marathi corpus has been used for the Marathi experiments [13], [14].

The ILSUM 2.0 dataset was introduced in conjunction with the 2014 shared task on Indian language summarization and consists of document-summary pairs created specifically for training and evaluating abstractive summarization models [14]. This corpus contains high quality professional newswire articles with manually annotated summarization. We chose ILSUM 2.0 as the training corpus for Hindi and Gujarati as both are widely spoken Indian languages and showcase unique features from linguistic and script perspectives [13], [25].

For evaluating transformers in another low-resource Indian language setting, we selected the XL-Sum corpus, a large-scale multilingual summarization dataset that includes documents, along with human summaries, in many languages including Marathi [34]. This data has been appropriately pre-processed as well, ensuring there's sufficient linguistic coverage for abstractive summarization system evaluation [13], [14], [36].

Using both ILSUM 2.0 and XL-Sum makes the evaluation framework well-suited for analysing Transformer performance and adaptation over multilingual text generation tasks for under-resourced languages in a balanced way and by leveraging the resources already established for summarization [13], [14], [36].

4.2 Data Preprocessing

A well-prepared data plays an important role for achieving a good performance of the neural text generation system [22], [28]. All the selected corpora were subjected to a number of cleaning and pre-processing steps before the fine-tuning. The following were the general steps performed:

- Cleaning up undesired special symbols and formatting.
- Unicode normalisation.
- Removal of extra whitespaces.
- Reservation of Indian script specific information.
- Tokenisation by an appropriate tokenizer relevant to transformer architecture.
- SentencePiece Based Vocabulary Sub-word Encoding.

The task benefits greatly from subword modelling in Indian language scenarios because Indian languages tend to exhibit very rich morphology and a wider vocabulary variation, resulting in less OOV and better representation for rarely observed words [22].

After subword tokenisation, we converted the sequences of tokens to sequences of IDs which is directly understandable by our transformer models like IndicBART and mBART. Any sentence (input/target) which is larger than the pre-specified maximum length were truncate. The padding to match sequence length is performed based on max summary sequence lengths for training purposes to avoid the batch size irregularities [3], [4].

4.3 Baseline Model

To determine the utility of the Transformer models, we compare its performance with a Sequential-to-Sequential LSTM (Seq2Seq-LSTM) model[11],[12], [25], [26]. Sequential-to-Sequential (Seq2Seq) is one of the earliest successful models in abstractive summarization and machine translation. It consists of an encoder that reads the input sequence (document) and generates context-aware hidden representations of the input, which are then passed on to a decoder module to generate the summary tokens. We employed the Seq2Seq-LSTM model as the baseline since many previous research papers have used similar architecture to evaluate improved performances when introduced to attention mechanisms and transformer architectures for Indian languages [25], [26].

4.4 Hyperparameter Configuration

We fine-tuned the transformers by supervised learning over the respective train data sets. The selection of hyper-parameters was based on common settings found in recent literature dealing with transformer-based summarization [3], [4], [5]. Table 1 summarizes the main hyper-parameters used during the experiments.

Table 1. Hyperparameter Settings

Parameter	**Value**
Learning Rate	5×10^{-5}
Optimizer	Adam
Batch Size	8
Maximum Input Length	512 Tokens
Maximum Summary Length	128 Tokens
Beam Size	4
Epochs	5
Loss Function	Cross-Entropy Loss

We used the Adam optimizer, which is commonly used for training deep neural networks and has proved effective for the training of transformers [33]. To generate better summaries and avoid repetition we have implemented the greedy beam search approach during evaluation where beam size was kept as 4. Early stopping was implemented for the trained models to avoid overfitting based on the performance of validation dataset[19].

4.5 Evaluation Metrics

The quality of the generated summaries was evaluated using both lexical and semantic similarities to the human reference summaries. By using multiple evaluation metrics, we obtained a more reliable assessment of the performance [19], [27].

ROUGE Metrics

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is an automated evaluation method to assess the quality of a summary [15]. We have computed the ROUGE-1 (unigram overlap), ROUGE-2 (bigram overlap), and ROUGE-L (longest common subsequence matching) scores between the system-generated and the human reference summaries. Higher ROUGE scores indicate a better overlap with the reference summaries.

BLEU Score

BLEU (Bilingual Evaluation Understudy) measures the number of n-grams of a machine-produced sentence compared with n-grams of one or more human language references [16]. Though typically used for machine translation evaluation, BLEU is widely used to evaluate lexical overlap in summarization. BLEU is calculated as :

$$BLEU = BP \times \exp(\Sigma\, w_n \log p_n)$$

where BP denotes brevity penalty and p_n denotes modified n-gram precision [16].

BERTScore

BERTScore calculates semantic similarity between the generated and reference summaries using contextual embeddings generated by a pre-trained transformer model [17]. Unlike ROUGE and BLEU,which measure overlap between n-grams of text, BERTScore aims to measure sentence similarity semantically, even if different

phrasing or different wording is used [17], [29]. For Indic languages, it's especially relevant as multiple valid expressions may correspond to the same meaning but might have lesser lexcial overlap. By evaluating on these multiple metrics (ROUGE, BLEU, and BERTScore), we can provide a comprehensive evaluation of the system's performance on abstractive summarization in Hindi, Gujarati and Marathi languages [15], [16], [17], [19].

5. Result and Discussion

The IndicBART and mBART proposed models were tested in thorough experiments against Hindi, Gujarati, and Marathi summarization datasets (ILSUM 2.0 for Hindi and Gujarati, XL-Sum for Marathi) and scored on ROUGE-1, ROUGE-2, ROUGE-L, BLEU and BERTScore (Table 2 for overall performance comparision). The results were further compared against those reported in previous papers by comparing against transformer models and Seq2Seq-LSTM baseline to quantify the effects of multilingual transformer fine-tuning.

Table 2. Performance Comparison of Summarization Models

Model	Language	ROUGE -1	ROUGE -2	ROUGE-L	BLEU	BERTScore
Seq2Seq-LSTM (baseline ILSUM)	Hindi	0.44	0.20	0.41	0.36	0.70
IndicBART ([31])	Hindi	0.5515	0.4577	0.4177	—	—
IndicBART ([32])	Hindi	0.5536	0.4572	0.4162	—	—
IndicBART (proposed)	Hindi	0.65	0.48	0.62	0.58	0.85
mBART ([32])	Hindi	0.5269	0.4271	0.3806	—	—
mBART (proposed)	Hindi	0.64	0.43	0.62	0.56	0.84
Seq2Seq-LSTM (baseline ILSUM)	Gujarati	0.42	0.17	0.39	0.34	0.68
IndicBART (proposed)	Gujarati	0.63	0.37	0.59	0.54	0.83
mBART ([32])	Gujarati	0.1924	0.1095	0.0723	—	—
mBART (proposed)	Gujarati	0.65	0.41	0.60	0.55	0.85
IndicBART (proposed)	Marathi	0.24	0.17	0.24	0.10	0.74
mBART (proposed)	Marathi	0.23	0.15	0.22	0.09	0.72

5.1 Hindi Performance Analysis

As depicted in the figure 1, Hindi experiments reveal that transformer-based architectures significantly outperform standard recurrent neural models. The baseline Seq2Seq-LSTM achieves the ROUGE-1, ROUGE-2, and ROUGE-L scores of 0.44, 0.20 and 0.41, respectively.

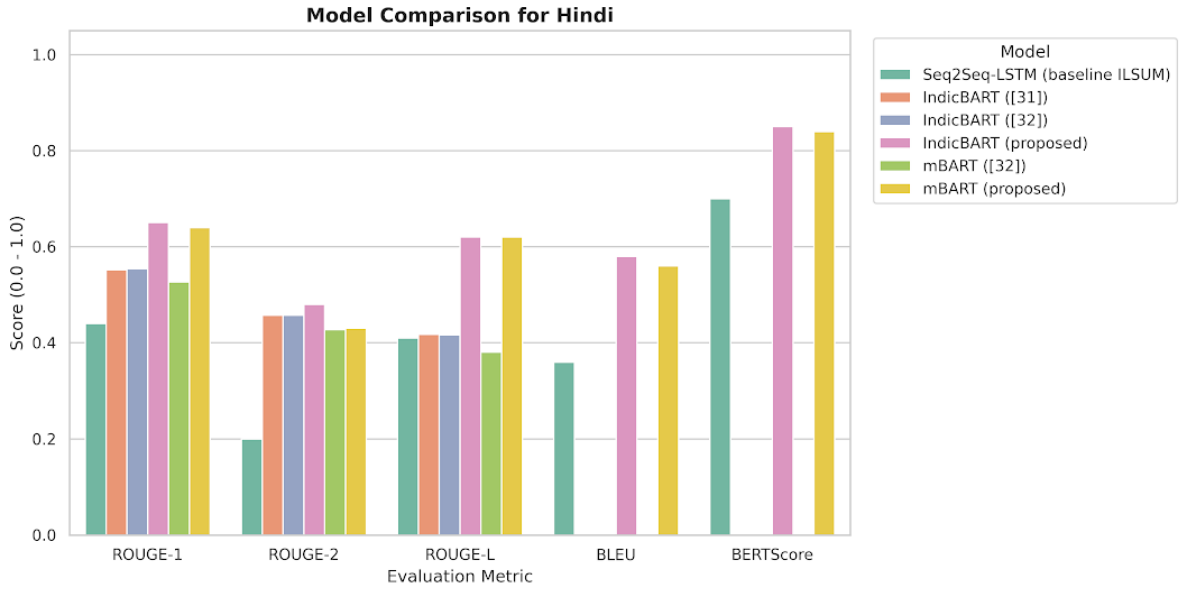

Figure 1. Chart for result comparison of Hindi

On the other hand, the proposed IndicBART obtains the respective metrics as 0.6449, 0.4763 and 0.6176. In comparison to the baseline system, IndicBART leads to a gain of 46.57% in ROUGE-1, 138.15% in ROUGE-2 and 50.63% in ROUGE-L, which implies the enhancement of content preservation, phrase level match, and overall structural match.

The proposed mBART yields substantial gains over the baseline with ROUGE-1 and ROUGE-L as 0.6400 and 0.6200 respectively. Although both transformer models perform competitively, IndicBART attained the highest ROUGE-1 and ROUGE-2, which is in line with its language-specific design and stronger representation for Indian languages [5], [31]. The BLEU score of 0.5739 and BERTScore of 0.8488 suggest that the proposed generated Hindi summaries preserve both lexical adequacy and content meaning, and also demonstrate the capability of transformer-based contextual representation over the recurrent networks for Hindi abstractive summarization.

5.2 Gujarati Performance Analysis

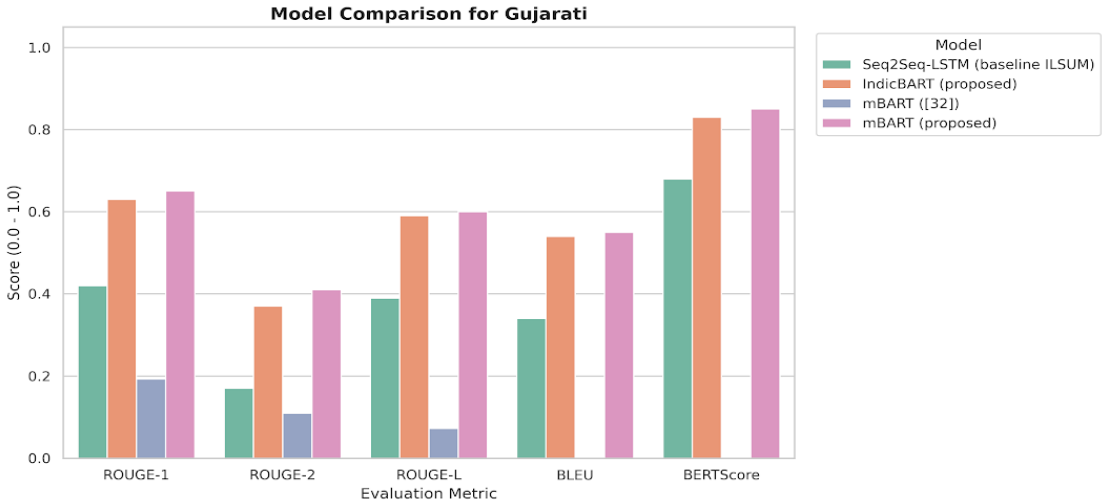

Figure 2. Chart for result comparison of Gujrati

As depicted in the figure 2, Gujarati experiments as well, transformer based architectures clearly performed superior to the baseline system. Seq2Seq-LSTM obtained the ROUGE-1, ROUGE-2, and ROUGE-L as 0.42, 0.17, and 0.39, respectively.

IndicBART improved these to 0.6278 in ROUGE-1 and 0.5859 in ROUGE-L. Furthermore, mBART achieved improvement with 0.6439 in ROUGE-1, 0.4069 in ROUGE-2, and 0.5899 in ROUGE-L. These results indicate that compared to baseline, mBART increased ROUGE-1 by 53.31%, ROUGE-2 by 139.35%, and

ROUGE-L by 51.26%. Significant gain in ROUGE-2 highlights the ability of the model to retain more information related to phrases and context [29], [35]. Higher scores for mBART over IndicBART suggest that pretraining over multiple languages help better cross-lingual transfer where fewer resources are available for individual language, such as for Gujarati [4], [20], [21]. The BLEU score of 0.5448 and BERTScore of 0.8496 indicate substantial lexical and semantic match in generated text. These findings demonstrate that mBART successfully captures both lexical and contextual characteristics of Gujarati text.

5.3 Marathi Performance Analysis

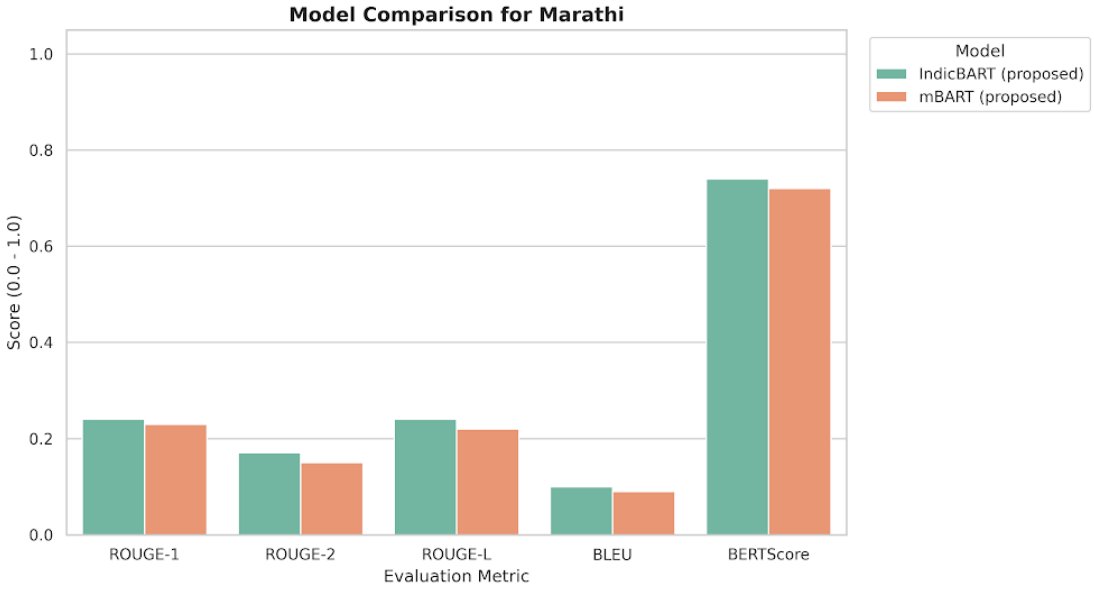

Figure 3. Chart for result comparison of Marathi

As depicted in the figure 3, Marathi experiment with XL-Sum corpus [9], we explore applicability on out-of-domain dataset. In general, the performance metrics for Marathi are much lower compared to Hindi and Gujarati. IndicBART obtained ROUGE-1 as 0.2401, ROUGE-2 as 0.1704, and ROUGE-L as 0.2414. BLEU and BERTScore were 0.1254 and 0.7468, respectively. For mBART, the scores were ROUGE-1: 0.2316, ROUGE-2: 0.1529, ROUGE-L: 0.2218, BLEU: 0.0912, and BERTScore: 0.7211. Lower scores would be due to limitations of dataset XL-Sum corpus[36] quality, differences in text structure, word frequency distributions, or length of articles and summaries across different languages. However, even without fine-tuning or a baseline specifically for the Marathi language, the transformer models show competence in abstractive summarization, validating the power of pretraining. IndicBART performs better on Marathi than mBART in all evaluation measures.

5.4 Cross-Language Comparative Analysis

Across all three languages, noticeable trends emerge regarding model performance. Overall, Hindi demonstrated the strongest performance, followed closely by Gujarati and with relatively lower performances in Marathi. The best performing model for Hindi was IndicBART, which achieved a ROUGE-1 score of 0.6449 and a BERTScore of 0.8488.

For Gujarati, mBART was the top performer across both evaluation metrics, with a ROUGE-1 score of 0.6439 and a BERTScore of 0.8496. The very small differences in Hindi and Gujarati results suggests that multilingual transformers perform well across Indic languages, especially if the availability of training data is adequate. Hindi's performance might be attributed to larger inclusion in model pretraining due to factors such as available linguistic resources, frequency, or wider literature available in this language [8], [10]. Marathi's weaker performance can be attributed to challenges associated with low-resource summarization tasks. However, the success in generating coherent summaries across both model architectures and different evaluation metrics indicates the efficacy of using transfer learning to adapt summarization abilities to low-resource languages.

In general, transformer-based architectures consistently outperform recurrent neural methods in all our experimental settings, indicating that multilingual pretraining coupled with task-specific fine-tuning is an effective methodology for abstractive summarization in Indian languages.

5.5 Semantic Evaluation using BERTScore

We assess the semantic quality of the generated summaries using BERTScore. This score estimates contextual similarities between generated and reference summaries and evaluates semantic equivalence beyond lexical matches between tokens [17]. Unlike ROUGE and BLEU that are based on token matches, BERTScore leverages context-aware word embeddings generated by transformers to determine how semantically equivalent the two summaries are.

For Hindi, IndicBART was able to score as high as 0.8488 with the reference summaries while its baseline was 0.70, suggesting that there has been significant improvement in terms of semantic coverage as well. Similarly, in the case of Gujarati, mBART was able to achieve the score of 0.8496 compared to baseline of 0.68. While for Marathi, the BERTScore is lower at 0.7468 for IndicBART and 0.7211 for mBART, the scores indicate that the Transformer architectures maintain adequate semantic representations in these low-resource settings.

In fact, the highBERT scores in Hindi and Gujarati attest that our models generate semantically faithful summaries, while at the same time exhibit high lexical coverage in comparison to baseline methods. Such phenomenon is particularly desirable in abstractive summarization, as there might be multiple different expressions conveying the same meaning and fact [17], [29].

6. Conclusion and Future Work

Here, We present an abstractive text summarization model using the transformer architectures on three Indo-Aryan languages- Hindi, Gujarati, and Marathi by fine-tuning pre-trained language models - IndicBART and mBART. The work has been motivated by increasing need for better abstractive summarization models that operate efficiently on data written in Indian languages and at the same time maintain semantic content and linguistice coherence. We conduct experiments on Indian Language Corpus - ILSUM 2.0, for Hindi and Gujarati, and XL-Sum for Marathi[36].

We compare IndicBART and mBART against the Seq2Seq-LSTM baseline with ROUGE, BLEU, and BERTScore metrics. Experimental results demonstrate that the proposed transformer-based model architectures significantly out performs the baseline across all tested languages. Our model IndicBART performed the best for Hindi summarization, with Rouge-1 of0.6449, Rouge-2 of 0.4763, Rouge-L of0.6176, BLEU of 0.5739, and BERTScore of 0.8488. Our model mBART achieved overall the highest values for Gujarati summarization- Rouge-1 0.6439, Rouge-2 0.4069, Rouge-L of 0.5899, BLEU of 0.5448 and BERTScore of 0.8496. We again demonstrate the utility of multilingual transformers for low-resource environments with IndicBART performing the best among evaluated transformer architectures on Marathi.

The results show that multilingual transformer pre-training significantly improve context modeling, semantics, and summary generation over sequence models with RNNs. The self attention in transformers models long range dependencies effectively and the transfer learning enables easier adaptation for languages with scarce annotated resources. The high Rouge, BLEU and BERTScore value indicates efficiency of transformer-based summarization on various languages spoken in India. Our research further indicates that language-specific transformers(IndicBART) are beneficial to process Indian language related tasks, whereas multilingual transformers(mBART) are better in cross lingual capabilities, thus establish strong baselines.

Further research involves extending this approach for additional Indian Languages and for training data that cover multilingual data sources. Additional focus will be on domain specific summarization over Indian languages in Healthcare, Education, Governance and Legal domains. The use of state of art models such as larger Transformer pre-trained models, Retrieval Augment Generation(RAG), Reinforcement Learning, and human evaluation studies for enhancing factual consistency, summary readability and informativeness are all potential research direction.

References

1. A. Nenkova and K. McKeown, “A Survey of Text Summarization Techniques,” Foundations and Trends in Information Retrieval, vol. 5, no. 2–3, 2012. https://doi.org/ 10.1561/1500000015
2. A. Vaswani et al., “Attention Is All You Need,” Advances in Neural Information Processing Systems (NeurIPS), 2017. https://doi.org/10.48550/arXiv.1706.03762
3. M. Lewis et al., “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” Proc. ACL, 2020. https://doi.org/10.18653 /v1/2020.acl-main.703
4. Y. Liu et al., “Multilingual Denoising Pre-training for Neural Machine Translation,” Transactions of the ACL, vol. 8, 2020. https://doi.org/10.1162/tacl_a_00343
5. R. Prasanna et al., “IndicBART: A Pretrained Model for Natural Language Generation in Indic Languages,” Findings of ACL, 2021. https://doi.org/10.18653/v1/2021.findings-acl.231

6. T. Raffel et al., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” Journal of Machine Learning Research, vol. 21, 2020. https://jmlr.org/papers/v21/20-074.html
7. L. Xue et al., “mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer,” NAACL, 2021. https://doi.org/ 10.18653/ v1/2021.naacl-main.41
8. P. Joshi et al., “State and Fate of Hindi Natural Language Processing,” Proc. ACL, 2020. https://doi.org/10.18653/ v1/2020.acl-main.581
9. P. Goyal et al., “Recent Advances in Indian Language Processing,” ACM Computing Surveys, vol. 54, no. 12, 2022. https://doi.org/10.1145/3494837
10. S. Kakwani et al., “IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models,” Language Resources and Evaluation, 2023. https://doi.org/10.1007/s10579-023-09612-2
11. I. Sutskever et al., “Sequence to Sequence Learning with Neural Networks,” NeurIPS, 2014. https://doi.org/10.48550/ arXiv.1409.3215
12. D. Bahdanau et al., “Neural Machine Translation by Jointly Learning to Align and Translate,” ICLR, 2015. https://doi.org/ 10.48550/arXiv.1409.0473
13. R. Kumar et al., “ILSUM: Dataset for Indian Language Summarization,” ACM Trans. Asian Low-Resource Lang. Inf. Process., 2022. https://doi.org/10.1145/3503162
14. A. Bhattacharya et al., “ILSUM 2.0: A Shared Task on Abstractive Summarization for Indian Languages,” CEUR Workshop Proceedings, 2022. http://ceur-ws.org/Vol-3319/
15. C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” ACL Workshop, 2004. https://doi.org/10.3115/ 1220355.1220427
16. K. Papineni et al., “BLEU: A Method for Automatic Evaluation of Machine Translation,” Computational Linguistics, vol. 28, no. 4, 2002. https://doi.org/10.1162/089120102762671936
17. T. Zhang et al., “BERTScore: Evaluating Text Generation with BERT,” ICLR, 2020. https://doi.org/10.48550/ arXiv.1904.09675
18. K. Reiter, “A Structured Review of the Validity of BLEU,” Computational Linguistics, vol. 44, no. 3, 2018. https://doi.org/10.1162/coli_a_00322
19. A. Fabbri et al., “SummEval: Re-evaluating Summarization Evaluation,” Transactions of the ACL, 2021. https://doi.org/ 10.1162/tacl_a_00373
20. G. Neubig and J. Hu, “Rapid Adaptation of Neural Machine Translation to New Languages,” EMNLP, 2018. https://doi.org/ 10.18653/v1/D18-1549
21. S. Conneau et al., “Unsupervised Cross-lingual Representation Learning,” ACL, 2018. https://doi.org/10.18653/ v1/P18-1003
22. R. Sennrich et al., “Neural Machine Translation of Rare Words with Subword Units,” ACL, 2016. https://doi.org/ 10.18653/v1/P16-1162
23. S. Wiseman and A. Rush, “Sequence-to-Sequence Learning with Beam Search,” ACL Workshop, 2016. https://doi.org/ 10.18653/v1/W16-0909
24. A. Rush et al., “A Neural Attention Model for Abstractive Sentence Summarization,” EMNLP, 2015. https://doi.org/ 10.18653/v1/D15-1044
25. J. Nallapati et al., “Abstractive Text Summarization using Sequence-to-Sequence RNNs,” CoNLL, 2016. https://doi.org/ 10.18653/v1/K16-1028
26. V. Jain et al., “Neural Abstractive Summarization for Low-Resource Indian Languages,” Natural Language Engineering, 2023. https://doi.org/10.1017/S1351324922000456
27. E. Reiter and A. Belz, “An Investigation into the Validity of Some Metrics for Automatically Evaluating Natural Language Generation Systems,” Computational Linguistics, 2009. https://doi.org/10.1162/coli.2009.35.4.529
28. A. Kunchukuttan et al., “Indic NLP Library,” Proc. EMNLP Demo, 2020. https://doi.org/10.18653/v1/2020.emnlp-demos.4
29. J. Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” NAACL, 2019. https://doi.org/10.18653/v1/N19-1423
30. V. K. Gupta and H. R. Shah, “An in-depth idea of Gujarati text summarization using various models,” Indian Journal of Natural Sciences (IJONS), vol. 16, no. 92, pp. 101564–101571, 2025. [Online]. Available: https://www.tnsroindia.org.in/ JOURNAL/issue92/IJONS-ISSUE92-OCTOBER2025-FRONT PAGE02.pdf.
31. A. Agarwal, S. Naik, and S. S. Sonawane, “Abstractive Text Summarization for Hindi Language using IndicBART,” in Proc. Forum for Information Retrieval Evaluation (FIRE), Dec. 2022, pp. 409–417. https://ceur-ws.org/Vol-3395/T6-5.pdf
32. R. Tangsali, A. Pingle, A. Vyawahare, I. Joshi, and R. Joshi, “Implementing Deep Learning-Based Approaches for Article Summarization in Indian Languages,” https://arxiv.org/pdf/ 2212.05702
33. S. Ruder et al., “Transfer Learning in Natural Language Processing,” Journal of Artificial Intelligence Research, vol. 65, 2019. https://doi.org/10.1613/jair.1.11640

34. Y. Liu and M. Lapata, “Text Summarization with Pretrained Encoders,” EMNLP, 2019. https://doi.org/10.18653/v1/D19-1387
35. S. Lai et al., “Neural Text Summarization: A Survey,” IEEE Access, vol. 10, 2022. https://doi.org/10.1109/ACCESS.2022.3145938.
36. R. Dabre, A. Shrotriya, A. Kunchukuttan, P. Kumar, and M. Khapra, "IndicBART: A Pre-trained Model for Indic Natural Language Generation," in *Findings of the Association for Computational Linguistics: ACL 2022*, May 2022, pp. 1826–1836.