Indian Language Summarization and Factual Error Detection using Pretrained Sequence-to-Sequence Models and Large Language Models

Abhijith Balan, Karthik Chittoor and Oswald Christopher

Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli

Abstract

Text summarization is an application Natural Language Processing where we shorten the text without losing essential information. Indian Language Text Summarization (ILTS) is a type of text summarization application focused on methods to shorten text in the context of Indian languages. It has many application such as document understanding, question answering, content curation etc., The main scope of the proposed work is to preserve Indian Languages and apply modern Machine Learning and Deep learning models for ILTS. The ILSUM shared task addresses text summarization and factual verification for multiple Indian languages alongside English. For Task 1, we leveraged pre-trained summarization and text-to-text generation models from Hugging Face, fine-tuning them on the provided dataset to generate accurate, concise summaries. For Task 2, we employed quantized model of LLAMA3 a Large Language Model(LLM) using Ollama platform, to identify factual incorrectness in cross-lingual machine-generated summaries. This paper provides a detailed overview of our proposed approach, including the models selected, fine-tuning techniques, and evaluation of results, along with an analysis of model effectiveness for each task.

Keywords

Indian Language Summarization, Pre-trained models, Fine-tuning, Text-to-text generation, Large Language Models (LLMs), Multilingual models, Factual error detection,

1. Introduction

Automatic text summarization is a process in natural language processing (NLP) that generates a concise and coherent version of a longer text. The goal is to retain essential information, enabling quicker understanding without reading the entire content. Summarization techniques are broadly categorized into extractive and abstractive methods. Extractive summarization selects key sentences from the original text, while abstractive summarization generates new sentences to convey the main ideas. This technology is widely used in applications like news aggregation, content curation, and document summarization for enhanced readability and efficiency. The progress of text summarization has, however, been hindered due to the lack of resources and high-quality datasets. Nevertheless, the availability of large-scale multilingual datasets such as XL-Sum [1] have led to substantial progress in natural language generation and summarization tasks. Even though quality-wise, these datasets are far from perfect, they do serve as a good starting point in terms of quantity. In the case of Indian Languages, there is still work yet to be done. The motivation behind to do shared task is to preserve Indian Languages and to design and fine tune Machine Learning and Deep Learning Model for summarization task.

Additionally, recent advancements in neural-based pretrained models have transformed the field significantly. In paper [2], discusses a Deep Learning based model named as Deep Learning Modifier Neural Network (DLMNN) where based on entropy values we classify most appropriate summarization for a given text. A graph based approach is used for text summarization wherein weights are assigned on edges consisting of sentences based sentence rank and similarity [3]. In paper [4], discusses automatic text summarization using SpaCy and Python. In some cases, domain knowledge base is used for text summarization[5]. The goal of the ILSUM shared task is to create reusable corpora for Indian language

Forum for Information Retrieval Evaluation, December 12-15, 2024, India

406123001@nitt.edu (A. Balan); 106121033@nitt.edu (K. Chittoor); oswald@nitt.edu (O. Christopher)

summarization. The dataset is created by scraping the news articles and corresponding descriptions from publicly available news websites. ILSUM data consists of a summarization corpus for five major Indian languages- Hindi, Telugu, Tamil, Kannada, Bengali and Gujarati, along with Indian English.

This paper provides a comprehensive overview of the pre-trained sequence-to-sequence models we experimented with for Indian language and English summarization. We explored models such as mT5, MBart, and IndicBART for Hindi and Gujarati, and fine-tuned PEGASUS, BART, T5, and ProphetNet on English data. However, for the final implementation, we used the mT5 Multilingual XL-Sum model for summary generation in Hindi, Telugu, Tamil, Bengali, and English. For Task 2, aimed at detecting factual inaccuracies in cross-lingual summaries, we implemented the Ollama large language model (LLM).

For Kannada, the mT5 Multilingual XL-Sum's tokenizer was not optimized for tokenizing Kannada. To address this, we translated the Kannada text to English, used the model to generate summaries in English, and then translated the summaries back to Kannada. This approach allowed us to effectively work around the limitations of the tokenizer while still utilizing the power of the mT5 model for summarization.

2. Related Work

Text summarization has evolved from early rule-based and extractive models, such as frequency-based methods and graph algorithms like TextRank[6], to advanced neural models leveraging Seq2Seq and attention mechanisms. Transformer-based architectures, especially with the introduction of pretrained models like BERT[7], BART[8], and T5[9], have significantly advanced both extractive and abstractive summarization. Modern models focus on ensuring faithfulness, improving adaptability across domains, and expanding into multimodal summarization to handle diverse content formats.

The Indian Language Summarization (ILSUM) shared task, introduced at FIRE 2022 used a news article-summary dataset in Hindi, Gujarati, and English[10]. In this shared task, pre-trained models such as MT5, MBart, and PEGASUS were used and evaluation were done using ROUGE metrics indicated better performance for Hindi and English tasks, while Gujarati presented challenges due to its linguistic diversity. Future work aims to expand the dataset to include Bengali, Tamil, and Telugu, further supporting the development of NLP in under-resourced languages[11].

In the ILSUM 2023 track at FIRE built on previous efforts by adding Bengali and introducing a subtask on misinformation detection for machine-generated summaries. Task 1 focused on summarizing diverse news articles, capped at 75 words, while Task 2 required participants to classify factual inaccuracies in summaries, identifying issues like misrepresentation and false attribution. Evaluation metrics included ROUGE, BERT-score, and macro-F1, highlighting the growing complexity of the summarization tasks and the need for more sophisticated evaluation methods[12].

Moreover, the paper "Fighting Fire with Fire: Adversarial Prompting to Generate a Misinformation Detection Dataset" presents a novel approach to building a misinformation dataset using adversarial prompts with large language models (LLMs) such as GPT-4. By generating misinformation-laden summaries, the authors aimed to tackle the limitations of traditional datasets, which are often resource-intensive to create. This innovative method produced four types of misinformation—fabrication, false attribution, inaccurate quantities, and biased misrepresentation—highlighting the potential for LLMs to enhance misinformation detection research[13].

While traditional metrics like ROUGE are widely used, new methods, including BERTScore, address limitations in semantic evaluation, marking a shift toward more meaningful assessments in summarization. This growing body of research indicates a promising direction for enhancing the quality and reliability of summarization systems, particularly in under-resourced languages where linguistic diversity presents unique challenges.

3. Corpus Description

The dataset for **Task 1 (Text Summarization for Indian Languages)** is constructed using article and headline pairs from several leading newspapers across the country. For each language (except Tamil), over 15,000 news articles are provided[16]. The goal is to generate meaningful, fixed-length summaries—either extractive or abstractive—for each article. While previous works in other languages have used article-headline pairs, this dataset presents unique challenges of code-mixing and script-mixing, as Indian language news articles frequently incorporate English phrases.

Task 2 (Detecting Factual Incorrectness in Machine-Generated Cross-Lingual Summaries) is a continuation from the previous edition. The dataset is of a cross-lingual setup where the source document is in English, but the target summaries are in Gujarati or Hindi. For the training set, participants are provided with the source document in English, along with summaries in English, Hindi, and Gujarati[16]. The test set includes the English source document and summaries in Hindi and Gujarati. The task is to identify factual errors in the machine-generated summaries, and each data point is categorized based on four types of factual errors.

4. Model Description

The pretrained language models (PLMs) used for downstream tasks are pretrained using massive amounts of unlabeled text data. A PLM encodes extensive linguistic knowledge into a vast amount of parameters, which stimulates universal representations and improves generation quality. We have experimented with various pretrained generation models to find the optimal architecture.

T5 [17]. model proposes defining every NLP task in a text-to-text format, utilizing an encoder-decoder Transformer architecture fine-tuned on the C4 corpus. In our experiments, we utilize both the T5-Base (220M parameters) and T5-Large (770M parameters) versions of the model. Since T5 is trained on an English-only dataset, we also explore the multilingual variants, particularly mT5, a multilingual extension of T5. The mT5 model is trained on 101 languages, leveraging the mC4 dataset and fine-tuned on the XLSUM dataset specifically for summarization tasks across various languages. Owing to the large size of the models, we only fine-tuned the base version (580M parameters) of the mT5 model, as the large version has 1.2 billion parameters. This design and its extensive pre-training enable mT5 to achieve state-of-the-art performance on multilingual benchmarks, making it a suitable choice for our summarization tasks. All code and model checkpoints used in our work are publicly available[18].

Language	ROUGE-1	ROUGE-2	ROUGE-L
Bengali	29.5653	12.1095	25.1315
English	37.601	15.1536	29.8817
Gujarati	21.9619	7.7417	19.86
Hindi	38.5882	16.8802	32.0132
Tamil	24.3326	11.0553	22.0741
Telugu	19.8571	7.0337	17.6101

Table 1
ROUGE Scores for Different Languages

LLAMA3 [19] is a large language model (LLM) that specializes in understanding and generating human-like text. It leverages state-of-the-art transformer architecture to perform various NLP tasks, including summarization, translation, and question-answering. LLAMA3 is designed to handle cross-lingual tasks effectively, allowing for seamless interaction between different languages. We observed that LLAMA3 was able to tokenize and understand all the desired Indian languages, unlike some models from Hugging Face, which prompted us to utilize it for detecting factual incorrectness in machine-generated summaries. Its capabilities make it an essential tool in our approach to Task 2 of the ILSUM shared task, enhancing the accuracy of factual verification across a range of languages. Table 2, 3 and 4 shows performance of LLAMA3.

Table 2
Meta Llama 3 Instruct Model Performance

Task	Meta Llama 3 70B	Gemini Pro 1.5	Claude 3 Sonnet
MMLU (5-shot)	82.0	81.9	79.0
GPQA (0-shot)	39.5	41.5	38.5
HumanEval (0-shot)	81.7	71.9	73.0
GSM-8K (8-shot, CoT)	93.0	91.7	92.3
MATH (Minerva prompt)	50.4	58.5	40.5

Table 3
Meta Llama 3 Pre-trained Model Performance

Task	Meta Llama 3 70B	Gemini Pro 1.0	Mixtral 8x22B
MMLU (5-shot)	79.5	71.8	77.7
AGIEval English (3-shot)	63.0	–	61.2
BIG-Bench Hard (3-shot)	81.3	75.0	79.2
ARC-Challenge (25-shot)	93.0	–	90.7
DROP (3-shot, F1)	79.7	74.1	77.6

Table 4
Meta Llama 3 Instruct Human Evaluation (Aggregated)

Comparison	Win (%)	Tie (%)	Loss (%)
Meta Llama 3 70B Instruct vs Claude Sonnet	52.9	12.9	34.2
Meta Llama 3 70B Instruct vs Mistral Medium	59.3	11.4	29.3
Meta Llama 3 70B Instruct vs GPT-3.5	63.2	9.7	27.1
Meta Llama 3 70B Instruct vs Meta Llama 2	63.7	13.9	22.4

5. Architecture

The control flow of the architecture for each task is outlined in the following sections, where we demonstrate the distinct processes involved in generating multilingual summaries and in identifying various types of incorrectness in summaries. Each task leverages different model structures and methodologies tailored to its specific goals. Task 1 focuses on using the mT5 XL Sum Model to create accurate and coherent summaries across multiple languages, while Task 2 employs the Llama 3 model with zero-shot prompting to analyze and categorize inaccuracies in summaries, such as factual, contextual, and linguistic errors.

5.1. Task 1: Generating Summaries in Multiple Languages using mT5 XL Sum Model

The architecture for generating multilingual summaries utilizes the **mT5 XL Sum Model** to produce concise summaries in various languages. First, multilingual input text undergoes a **Preprocessing** stage, where it is tokenized and normalized to ensure compatibility with the model. This prepared text is then fed into the **mT5 XL Sum Model**, a powerful multilingual Transformer model fine-tuned specifically for summarization tasks. After the model generates the summary, a **Postprocessing** stage formats and refines the output, addressing any language-specific adjustments needed to ensure clarity and coherence. The result is a multilingual **Output** summary tailored to the language and content of the input text, facilitating cross-lingual summarization.

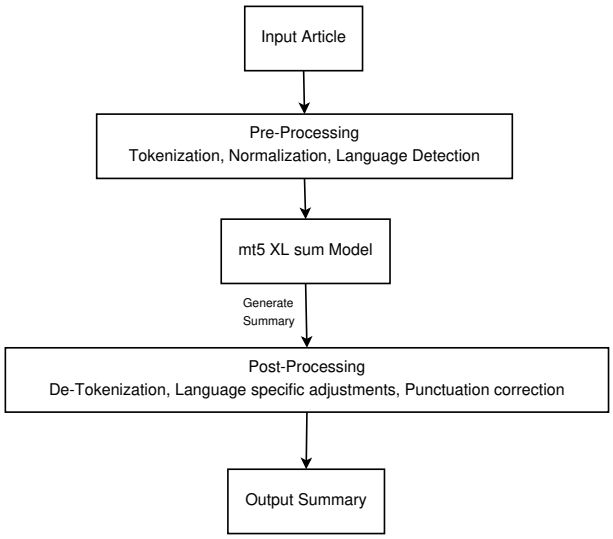

Figure 1: Architecture for Task 1: Generating multilingual summaries using mT5 XL Sum Model.

5.2. Task 2: Identifying Types of Incorrectness in Summaries using Llama 3 with Zero-Shot Prompting

In this architecture, the **Llama 3 model** is employed with zero-shot prompting to identify types of incorrectness in summaries. The process begins with the **Input** of an article and its generated summary, which undergoes an optional **Preprocessing** stage for text cleaning or formatting. Next, the **Llama 3 Model** receives zero-shot prompts instructing it to evaluate the summary against the article, identifying factual, contextual, and linguistic discrepancies. After processing, a **Postprocessing** step interprets and structures the model's output. The identified discrepancies are then classified in the **Error Type Classification** stage into factual, contextual, or linguistic errors, providing a categorized **Output** report that highlights areas where the summary deviates from the intended information in the original article.

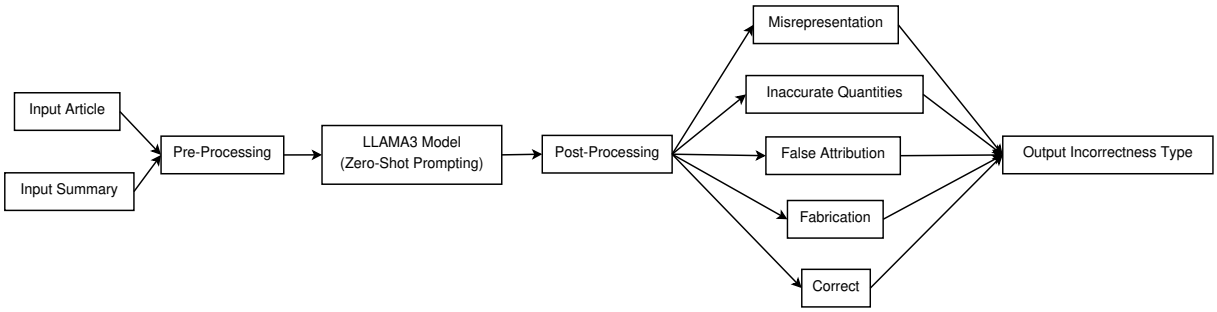

Figure 2: Architecture for Task 2: Identifying types of incorrectness in summaries using Llama 3 with zero-shot prompting.

6. Training

For Task 1, we trained our summarization model on each language dataset for one epoch due to time constraints. Here's a detailed breakdown of the training parameters and libraries used:

6.1. Libraries Used

- **Transformers:** The Hugging Face transformers library allowed us to fine-tune large, pre-trained models for summarization, providing an accessible, high-level API for handling the complexity of sequence-to-sequence models.

- **SentencePiece:** Essential for tokenization, SentencePiece enabled efficient vocabulary management across multiple languages, which was especially helpful with diverse multilingual data.
- **Torch (PyTorch):** Used as the base deep learning framework, PyTorch allowed for efficient handling of data loading, GPU allocation, and batch processing during training.

6.2. Training Parameters

- **Batch Sizes:**
  - To handle memory constraints, `per_device_train_batch_size` and `per_device_eval_batch_size` were set to 1. This minimized memory usage on each GPU.
  - **Gradient Accumulation:** By setting `gradient_accumulation_steps` to 16, we created an effective batch size by accumulating gradients across 16 steps before each model weight update. This approach allowed us to keep memory usage low while maintaining training stability.
- **Warmup Steps:**
  - `Warmup Steps (500):` A gradual increase in the learning rate during the initial phase of training stabilized training and improved convergence, especially for the complex, multilingual datasets.
- **Weight Decay:**
  - With 0.01 as the weight decay parameter, the model applied regularization to prevent overfitting by discouraging overly large weights.
- **Evaluation and Logging Strategy:**
  - **Evaluation Strategy:** `evaluation_strategy='steps'` enabled regular evaluation during training, providing ongoing feedback on model performance. We evaluated every 500 steps (`eval_steps=500`) to track metrics and adjust as needed.
  - **Logging Frequency:** Logging every 10 steps (`logging_steps=10`) gave us detailed insight into training progress and allowed us to quickly identify potential issues.
- **Model Saving Strategy:**
  - With `save_steps` set to 1e6, we minimized save checkpoints to avoid interruptions and maintain efficient training flow.

6.3. Memory Management

Memory constraints were a significant factor, so we optimized the configuration with small batch sizes and gradient accumulation. This approach allowed us to train effectively within the available resources, balancing memory usage with training effectiveness.

For Task 2, we applied zero-shot classification using the LLAMA3 language model without any additional training. In this approach, the model leveraged its existing knowledge to perform classification directly on machine-generated summaries, identifying factual inaccuracies across different categories without the need for fine-tuning.

7. Evaluation Metrics

ROUGE Scores: The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric is primarily used for automatic summarization evaluation. It measures the overlap between the generated summaries and reference summaries by calculating n-gram recall. Key variants include ROUGE-1 (unigrams), ROUGE-2 (bigrams), ROUGE-4 (four-grams), and ROUGE-L (longest common subsequence), which capture different aspects of summary quality. Table 5 shows the ROUGE score of our proposed work for task 1.

- **ROUGE-N**: Measures the recall of n-grams between the generated and reference summaries. For ROUGE-1, ROUGE-2, and ROUGE-4, the formula is:

$$\text{ROUGE-N} = \frac{\sum_{\text{ref} \in \text{references}} \sum_{\text{gram}_n \in \text{ref}} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{\text{ref} \in \text{references}} \sum_{\text{gram}_n \in \text{ref}} \text{Count}(\text{gram}_n)}$$

  where $\text{Count}_{\text{match}}(\text{gram}_n)$ is the number of n-grams in the generated summary that match the reference summary.
- **ROUGE-L**: Measures the longest common subsequence (LCS) between the generated and reference summaries. The ROUGE-L formula is:

$$\text{ROUGE-L} = \frac{\text{LCS}(\text{generated}, \text{reference})}{\text{Length of Reference}}$$

BERT Scores: BERT (Bidirectional Encoder Representations from Transformers) scores assess the performance of text classification tasks by evaluating model precision, recall, and F1 score. These metrics are based on the outputs of a fine-tuned BERT model, measuring how well the model identifies relevant information in the text. Table 6 shows the BERT score of our proposed work for task 1.

- **Precision:** The fraction of relevant instances among the retrieved instances:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Recall:** The fraction of relevant instances that were retrieved:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **F1 Score:** The harmonic mean of precision and recall:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1 Scores: The F1 score is the harmonic mean of precision and recall, providing a balance between the two. It is particularly useful in situations where the class distribution is imbalanced, as it emphasizes the importance of both correctly identifying positive instances and minimizing false positives. Table 7 shows the F1 score of our proposed work for task 2. The F1 score formula is:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

7.1. Task 1: ROUGE Scores

The ROUGE-L scores in Table 5 reveal varied performance across languages. Hindi achieved the highest score of 0.2981, indicating strong coherence in its generated summaries. Gujarati and English followed with scores of 0.2607 and 0.2497, reflecting good similarity to reference texts. In contrast, Kannada had the lowest score of 0.0491, likely due to challenges in generating coherent summaries, as the text was translated to English, summarized, and then translated back to Kannada. This analysis highlights the effectiveness of different language models in summarization tasks, suggesting that performance may be influenced by the quality of training data and linguistic characteristics.

Language	Rank	ROUGE-1	ROUGE-2	ROUGE-4	ROUGE-L
Bengali	2	0.2411	0.1610	0.1133	0.2233
English	6	0.2989	0.1392	0.0927	0.2497
Gujarati	2	0.2723	0.1467	0.0860	0.2607
Hindi	3	0.3310	0.1630	0.0978	0.2981
Kannada	2	0.0524	0.0095	0.0012	0.0491
Tamil	3	0.1962	0.1175	0.0801	0.1872
Telugu	3	0.1973	0.1161	0.0755	0.1916

Table 5
ROUGE Scores for Different Languages in Task 1

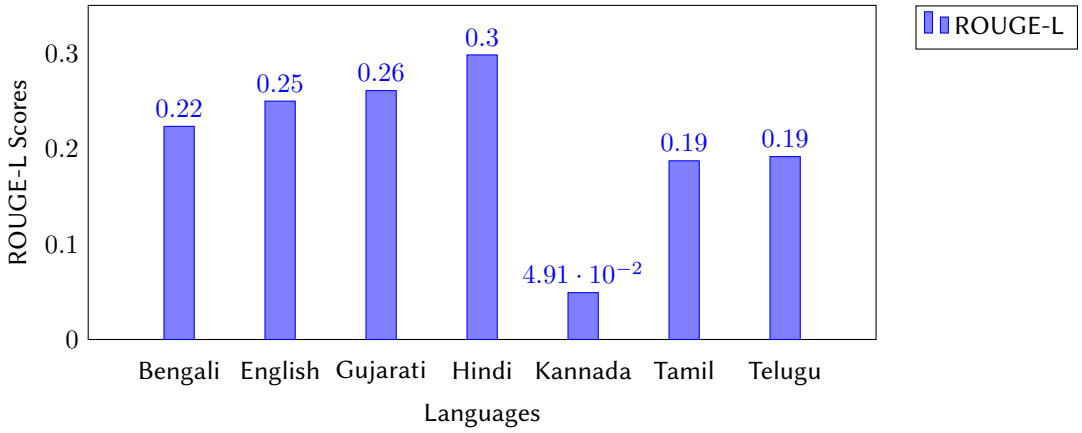

Figure 3: ROUGE-L Scores for Different Languages

7.2. Task 1: BERT Scores

The BERT scores in Table 6 and Figure 2 reveal notable performance across languages in terms of F1 scores. English leads with an F1 score of 0.8672, indicating the model's effectiveness with English text. Bengali, Gujarati, and Hindi also perform well, with scores between 0.7310 and 0.7488, suggesting that BERT effectively captures their linguistic nuances. Conversely, Kannada's lower F1 score of 0.6691 indicates potential limitations in dataset quality or model performance. Overall, these results highlight BERT's strengths in more widely used languages and the need for improvements in less prevalent ones.

Language	Rank	Precision	Recall	F1 Score
Bengali	2	0.7384	0.7251	0.7310
English	4	0.8684	0.8664	0.8672
Gujarati	2	0.7485	0.7303	0.7388
Hindi	2	0.7410	0.7343	0.7371
Kannada	2	0.6713	0.6677	0.6691
Tamil	3	0.7123	0.7283	0.7197
Telugu	3	0.7175	0.7164	0.7166

Table 6
BERT Scores for Different Languages in Task 1

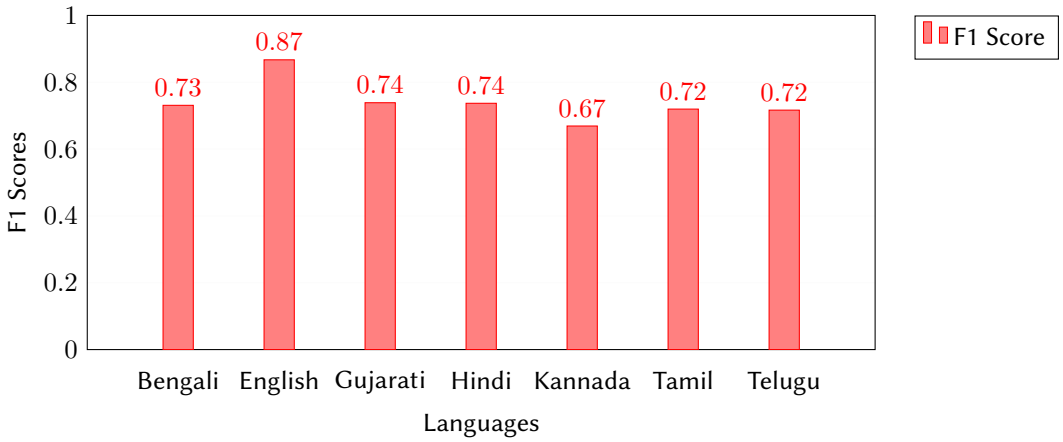

Figure 4: F1 Scores for Different Languages

Note on Kannada Scores: The ROUGE and BERT scores for Kannada are significantly lower compared to other languages. This is primarily due to the mT5 model we chose, as its tokenizer is not designed to effectively tokenize Kannada text, leading to poor comprehension of the language. To address this issue, we utilized a different translation model: we first translated the Kannada text to English, performed the summarization in English, and then translated the summary back to Kannada.

7.3. Task 2: F1 Scores

The F1 scores for Task 2, shown in Table 7 and Figure 3, indicate low performance for both Gujarati (0.1892) and Hindi (0.1810). These scores reflect challenges in producing high-quality outputs for these languages. The limited performance may stem from insufficient training data or the model's difficulty in capturing their linguistic nuances. There is significant potential for improvement in summarization techniques for Gujarati and Hindi, possibly by utilizing larger datasets or refining model architectures specifically for these languages. **Note on Task 2 Methodology:** For Task 2, we did not train the

Language	Rank	F1 Score
Gujarati	5	0.1892
Hindi	7	0.1810

Table 7
F1 Scores for Different Languages in Task 2

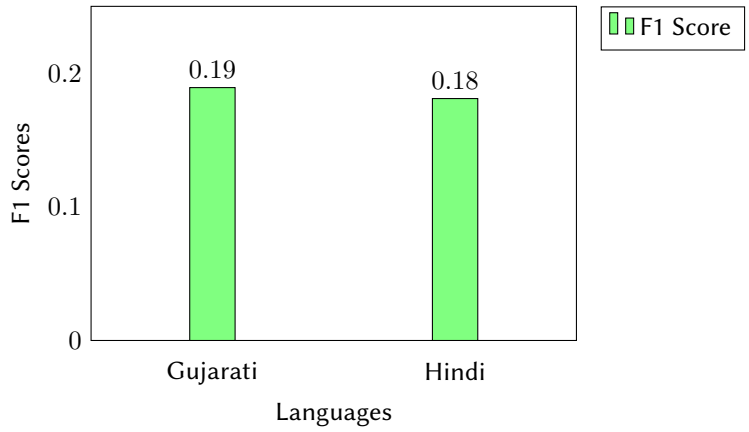

Figure 5: F1 Scores for Gujarati and Hindi

LLAMA3 model due to limited resources, such as the availability of faster GPUs and time constraints. Instead, we employed a zero-shot classification approach to evaluate the F1 scores for Gujarati and Hindi. This lack of extensive training likely contributed to the lower F1 scores observed for these languages.

8. Discussion and Conclusions

While having better models fine-tuned exclusively on Indian languages might benefit research in the area of Indian Language Summarization, creating larger, high-quality datasets for such languages will surely lead to progress in this field [18]. Many Indian languages have limited resources available for training robust NLP models, which poses a challenge to the development of effective summarization systems [20]. We conclude that the transformer-based pretrained LLMs and seq2seq models are capable of generating high-quality summaries for the ILSUM shared task. These models leverage the strengths of deep learning architectures to understand and summarize textual information effectively. However, our current results for Task 1 are limited due to the fact that we only trained our model for one epoch. This short training duration likely contributed to the lower scores observed, indicating that further improvements can be achieved with extended training sessions. Given more time and computational resources, we believe that additional epochs would enable the model to learn more complex patterns and nuances within the data, ultimately leading to better summarization performance.

9. Future Work

Looking ahead, we plan to train indicBART for Task 1, as it has shown to be more efficient according to last year's winners [20]. indicBART is specifically designed for Indian languages and has the potential to address some of the limitations observed in previous models [17]. Its architecture enables better handling of multilingual inputs and allows for fine-tuning on specific tasks, which can significantly improve summarization quality [21].

In addition, for Task 2, we utilized text classification models to detect types of incorrectness in the generated summaries corresponding to the articles. Unlike large language models, which can be resource-intensive and complex, text classifiers are more efficient for this specific task. By employing models like BERT or RoBERTa fine-tuned for binary or multi-class classification tasks, we can systematically evaluate the quality of summaries based on specific criteria such as factual accuracy, coherence, and fluency [13]. This approach allows us to identify and categorize various errors in the summaries, which can guide further refinements in our summarization systems.

Furthermore, we aim to conduct comparative studies to assess the performance of indicBART against other state-of-the-art models [10]. By evaluating its performance across various Indian languages and datasets, we hope to uncover insights into the strengths and weaknesses of different summarization strategies. This comprehensive analysis will not only contribute to the body of knowledge in the field but also inform future directions for research in Indian language summarization, ensuring that we continually advance the state of the art in this important area.

Acknowledgments

We thank the organizers of the ILSUM shared task for their help and support.

References

[1] T. Hasan, A. Nusrat, M. S. Sakib, M. S. I. Bhuiyan, N. M. T. Khan, M. R. Kamal, M. S. Shahriar, M. N. Rahman, Xl-sum dataset, 2021. URL: https://huggingface.co/datasets/csebuetnlp/xlsum.

[2] B. Muthu, S. Cb, P. M. Kumar, S. N. Kadry, C.-H. Hsu, O. Sanjuan, R. G. Crespo, A framework for extractive text summarization based on deep learning modified neural network classifier, ACM Trans. Asian Low-Resour. Lang. Inf. Process. 20 (2021). URL: https://doi.org/10.1145/3392048. doi:10.1145/3392048.

[3] S. Ullah, A. A. A. Islam, A framework for extractive text summarization using semantic graph based approach, in: Proceedings of the 6th international conference on networking, systems and security, 2019, pp. 48–56.

[4] S. JUGRAN, A. KUMAR, B. S. TYAGI, V. ANAND, Extractive automatic text summarization using spacy in python nlp, in: 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021, pp. 582–585. doi:10.1109/ICACITE51222.2021.9404712.

[5] P. Bhattacharya, S. Poddar, K. Rudra, K. Ghosh, S. Ghosh, Incorporating domain knowledge for extractive summarization of legal case documents, in: Proceedings of the eighteenth international conference on artificial intelligence and law, 2021, pp. 22–31.

[6] M. Zhang, X. Li, S. Yue, L. Yang, An empirical study of textrank for keyword extraction, IEEE Access 8 (2020) 178849–178858. doi:10.1109/ACCESS.2020.3027567.

[7] M. V. Koroteev, Bert: A review of applications in natural language processing and understanding, 2021. URL: https://arxiv.org/abs/2103.11943. arXiv:2103.11943.

[8] L. Cui, Y. Wu, J. Liu, S. Yang, Y. Zhang, Template-based named entity recognition using bart, 2021. URL: https://arxiv.org/abs/2106.01760. arXiv:2106.01760.

[9] A. Virani, R. Yadav, P. Sonawane, S. Jawale, Automatic question answer generation using t5 and nlp, in: 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), 2023, pp. 1667–1673. doi:10.1109/ICSCSS57650.2023.10169726.

[10] S. Satapara, B. Modha, S. Modha, P. Mehta, Findings of the first shared task on indian language summarization (ILSUM): approaches challenges and the path ahead, in: K. Ghosh, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, Kolkata, India, December 9-13, 2022, volume 3395 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 369–382. URL: https://ceur-ws.org/Vol-3395/T6-1.pdf.

[11] S. Satapara, B. Modha, S. Modha, P. Mehta, FIRE 2022 ILSUM track: Indian language summarization, in: D. Ganguly, S. Gangopadhyay, M. Mitra, P. Majumder (Eds.), Proceedings of the 14th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2022, Kolkata, India, December 9-13, 2022, ACM, 2022, pp. 8–11. URL: https://doi.org/10.1145/3574318.3574328. doi:10.1145/3574318.3574328.

[12] S. Satapara, P. Mehta, S. Modha, D. Ganguly, Indian language summarization at FIRE 2023, in: D. Ganguly, S. Majumdar, B. Mitra, P. Gupta, S. Gangopadhyay, P. Majumder (Eds.), Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2023, Panjim, India, December 15-18, 2023, ACM, 2023, pp. 27–29. URL: https://doi.org/10.1145/3632754.3634662. doi:10.1145/3632754.3634662.

[13] S. Satapara, P. Mehta, D. Ganguly, S. Modha, Fighting fire with fire: Adversarial prompting to generate a misinformation detection dataset, CoRR abs/2401.04481 (2024). URL: https://doi.org/10.48550/arXiv.2401.04481. doi:10.48550/ARXIV.2401.04481. arXiv:2401.04481.

[14] S. Satapara, P. Mehta, S. Modha, A. Hegde, S. HL, D. Ganguly, Overview of the Third Shared Task on Indian Language Summarization (ILSUM 2024), in: K. Ghosh, T. Mandl, P. Majumder, D. Ganguly (Eds.), Working Notes of FIRE 2024 - Forum for Information Retrieval Evaluation, Gandhinagar, India. December 12-15, 2024, CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[15] S. Satapara, P. Mehta, S. Modha, A. Hegde, S. HL, D. Ganguly, Key insights from the third ilsum track at fire 2024, in: Proceedings of the 16th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2024, Gandhiinagar, India. December 12-15, 2024, ACM, 2024.

[16] Indian Language Summarization (ILSUM), Ilsum 2024 database, 2024. URL: https://ilsum.github.io/ilsum/2024/Database.html, accessed: 2024-10-26.

[17] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Mubasshir, Y.-F. Li, Y.-B. Kang, M. S. Rahman, R. Shahriyar, XL-sum: Large-scale multilingual abstractive summarization for 44 languages, in: Findings of

the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 4693–4703. URL: https://aclanthology.org/2021.findings-acl.413.

[18] C. Karthik, ILSUM-FIRE, 2024. URL: https://github.com/ChittoorKarthik/ILSUM-FIRE/,https://huggingface.co/csebuetnlp/mT5_multilingual_XLSum, GitHub Repository, HugginFace Model.

[19] Meta, Introducing llama 3: Advancing open foundation models for generative ai, 2024. URL: https://ai.meta.com/blog/meta-llama-3/, meta AI Blog.

[20] A. Urlana, S. M. Bhatt, N. Surange, M. Shrivastava, Indian language summarization using pretrained sequence-to-sequence models, arXiv preprint arXiv:2303.14461 (2023).

[21] T. Hasan, A. Nusrat, M. S. Sakib, M. S. I. Bhuiyan, N. M. T. Khan, M. R. Kamal, M. S. Shahriar, M. N. Rahman, mt5 model for multilingual summarization - xl-sum, 2021. URL: https://huggingface.co/csebuetnlp/mT5_multilingual_XLSum.