Overcoming Code-Mixing and Script-Mixing in Indian Language Summarization with Transformer Models

Pulkit Chatwal[1,†], Amit Agarwal[2,†] and Ankush Mittal[3]

[1]Rajiv Gandhi Institute of Petroleum Technology, Jais, India
[2]AICoE Wells Fargo International Solutions Private Limited, Bangalore, India
[3]COER University, Roorkee India

Abstract

With the growing need for accessible information across diverse linguistic backgrounds, text summarization in multilingual contexts has become increasingly essential. Text summarization is a crucial task in natural language processing, particularly in multilingual settings involving Indian languages. This paper presents our approach for the FIRE 2024 task, where we leverage large transformer-based language models for summarizing Indian languages, addressing the linguistic diversity and frequent instances of code-mixing and script-mixing unique to this context. Our methodology incorporates both extractive and abstractive summarization techniques, optimized for Indian languages through advanced fine-tuning of models like mT5, IndicBART, and BART. While prompt engineering has predominantly been applied to English tasks, we adapt it alongside fine-tuning to enhance summarization performance and computational efficiency. Our models achieved top results in five languages—Hindi, Gujarati, English, Tamil, and Bengali—and ranked second in Telugu. These results demonstrate substantial improvements in summarization accuracy, underscoring our approach's efficacy in handling the complexities of Indian languages and advancing text processing in multilingual, mixed-language environments.

Keywords

Indian Languages, Text Summarization, Pre-Trained Model, Sequence-to-Sequence models, Multilingual Text Summarization, Transformer Models, Fine-Tuning, LLM

1. Introduction

With the vast amount of information generated daily across multiple languages, the ability to automatically summarize text has become essential for efficient information consumption and accessibility. In multilingual and multicultural societies like India, where linguistic diversity includes hundreds of languages and dialects, automatic summarization solutions are crucial for bridging communication gaps and ensuring equitable access to information. However, these challenges are further intensified by the prevalence of code-mixing (the blending of two or more languages within a single text) and script-mixing (the use of multiple writing systems), making conventional summarization methods insufficient. Addressing these complexities is essential for developing summarization tools that serve diverse user groups and linguistic contexts effectively.

Traditional summarization approaches often fall short in managing these complexities, particularly in multilingual settings. For instance, [1] highlights the limitations of conventional summarization techniques when dealing with code-mixed text. They emphasize the need for advanced, automatic summarization methods capable of processing complex, multi-modal data, such as text, images, and audio, to meet strategic information needs. Similarly, [2] explore the challenges faced by multilingual users who frequently engage in code-mixing and underscore the necessity for conversational agents designed to process mixed-language content effectively.

To address these challenges, we propose a dual approach that leverages both fine-tuning and prompt-based techniques applied to transformer-based models. Our solution involves fine-tuning multilingual models (mT5 and IndicBART) and English models (BART) on a diverse dataset of Indian languages to capture unique linguistic patterns, including code-mixing and script-mixing. Additionally, we employ

Forum for Information Retrieval Evaluation, December 12-15, 2024, India
[†]These authors contributed equally.
Pulkitchatwal@gmail.com (P. Chatwal); aagarwal3@cs.iitr.ac.in (A. Agarwal); dr.ankush.mittal@gmail.com (A. Mittal)

prompt engineering techniques to optimize summarization performance and computational efficiency, particularly for English-language content. This combined approach creates a robust, adaptable solution capable of generating accurate, concise summaries across a spectrum of Indian languages and mixed-language inputs.

2. Related Work

Recent advancements in text summarization have employed a range of methods and datasets, particularly focusing on fine-tuning transformer-based models for improved summarization performance. [3] introduced WikiLingua, a multilingual dataset with article-summary pairs in 18 languages, and fine-tuned the mBART model on this dataset. While effective, these efforts mainly focused on language pairs and did not address the complexities associated with code-mixing and script-mixing, both of which are crucial for multilingual contexts such as Indian languages. [4] utilized transformer-based models like RoBERTa-Base and Flan T5 Base for cross-platform age classification on social media, achieving impressive accuracy. [5] explored leadership traits during natural hazards by analyzing personality and emotional characteristics, uncovering key differences between local and global leaders. [6] introduced AgriLLM, leveraging transformers to automate query resolution for farmers and bridge information gaps in agriculture. [7] developed an SMS-based FAQ retrieval system using machine learning to refine noisy text and improve information access. Similarly, [8] fine-tuned T5, BART, and Pegasus for abstractive summarization of medical documents using the SUMPUBMED dataset, but their research did not extend to the multilingual, mixed-language contexts typical of Indian languages.

In the domain of extractive summarization, [9] proposed a graph-based approach that transformed text into a network of interconnected sentences and used a selectivity measure to assess node significance. This graph-based model, while innovative, is limited in handling language-specific nuances and lacks adaptability for abstractive summarization in mixed-language texts. [10] provided a comparative analysis of major extractive summarization techniques such as TF-IDF, Clustering, Fuzzy Logic, Neural Networks, and Graph-based methods, yet these methods typically struggle with capturing abstracted meaning, particularly in mixed-language settings.

For English text summarization, [11] demonstrated the effectiveness of BERT-based models across various datasets, showcasing strong summarization quality. Similarly, [12] explored reinforcement learning for abstractive summarization, optimizing both readability and content fidelity. However, these methods were focused on monolingual English text and did not address the unique challenges of Indian languages, where code-mixing and script-mixing are common.

In language-specific research, [13] used TextRank and Fuzzy C-means for Bengali text summarization, highlighting the importance of customized models for individual languages. For Gujarati, [14] combined pre-trained language models with clustering techniques, showing promising results for low-resource languages. However, these methods were largely extractive and were limited in their ability to handle code-mixed content across multiple Indian languages.

Studies from shared tasks organized by FIRE ([15]; [16]; [17]; [18]; [19]; [20]) have examined summarization challenges in Indian languages, employing diverse methodologies and models. Although these contributions have advanced summarization for Indian languages, they primarily focused on single-language settings or basic multilingual scenarios, lacking robust solutions for complex, code-mixed inputs.

Our Approach: Our work differs from previous studies by specifically targeting the multilingual, code-mixed, and script-mixed text summarization challenges inherent in Indian languages. We adopt a dual approach that combines fine-tuning and prompt-based methods for transformer-based models, such as mT5, IndicBART, and BART. By fine-tuning these models on a dataset of Indian languages and leveraging prompt engineering for computational efficiency, our approach is designed to capture the unique linguistic patterns of each language, including mixed-language constructs. This combined methodology allows us to generate coherent, high-quality summaries that address the specific complexities of Indian language contexts, setting our work apart as a comprehensive solution for multilingual summarization in code-mixed settings.

3. Problem Statement

Let D represent the dataset of news articles across multiple Indian languages, where:

$$D = \{(X_i, Y_i) \mid X_i \in \mathbb{X}, Y_i \in \mathbb{Y}\}_{i=1}^{N}$$

where: - X_i is the i-th article in the dataset, containing a mixture of text from multiple languages and potential code-mixing and script-mixing. - Y_i is the corresponding reference summary for X_i. - $\mathbb{X}$ is the space of input texts (articles), and $\mathbb{Y}$ is the space of target summaries. - N is the total number of article-summary pairs in the dataset.

The goal is to learn a mapping function $f : \mathbb{X} \to \mathbb{Y}$ that generates concise and informative summaries for each article X_i such that the generated summary $\hat{Y}_i = f(X_i)$ approximates Y_i.

3.1. Problem Objective

The objective is to minimize the error between the generated summaries $\hat{Y}_i$ and the reference summaries Y_i, typically measured using evaluation metrics such as ROUGE, BLEU, or cosine similarity in embedding space. Formally:

$$\min_f \sum_{i=1}^{N} \mathscr{L}(f(X_i), Y_i)$$

where: - $\mathscr{L}$ is the loss function representing the error between the generated summary $\hat{Y}_i = f(X_i)$ and the reference summary Y_i.

3.2. Additional Constraints and Considerations

- **Multilingual and Code-Mixed Text**: X_i may contain tokens from multiple languages $L = \{L_1, L_2, \dots, L_k\}$, where each L_j corresponds to a distinct language script. Thus, the model f must handle cross-lingual transfer effectively.
- **Length Constraint**: Each generated summary $\hat{Y}_i$ should ideally satisfy a fixed length constraint $|\hat{Y}_i| \approx K$, where K is the desired summary length.
- **Semantic Fidelity**: The mapping f should retain the essential semantic information from X_i in $\hat{Y}_i$, aligning with the reference summaries Y_i in terms of main facts and insights.

4. Model Description

English Language Model: BART For English-language summarization, we employ the facebook/bart-large-cnn model, a state-of-the-art transformer-based encoder-decoder (seq2seq) architecture. BART combines a bidirectional encoder, akin to BERT, with an autoregressive decoder, similar to GPT. This model is pre-trained with a denoising autoencoder objective, where it learns to reconstruct text corrupted by noise functions. This pre-training process equips BART to handle various downstream tasks, including summarization and translation, as well as comprehension tasks like text classification and question answering. For our experiments, we fine-tuned the model on the CNN/Daily Mail dataset, which contains a substantial collection of paired text and summary samples, ensuring robust performance in text summarization tasks [21].

Multilingual Language Models: mT5 and IndicBART To address summarization for multiple Indian languages, such as Gujarati, Telugu, and Bengali, we leverage the csebuetnlp/mT5_multilingual_XLSum model. mT5 is a multilingual variant of T5, designed to handle diverse languages by using a shared vocabulary and multilingual training data. For this task, we utilize XL-Sum, a high-quality, multilingual dataset curated for abstractive summarization with approximately 1 million article-summary pairs sourced from BBC News across 44 languages, including low-resource languages. XL-Sum emphasizes abstractive summarization with a high level of brevity, abstraction, and quality, as indicated by both human judgments and intrinsic metrics [22].

Hindi and Tamil Language Model: IndicBART For Hindi and Tamil summarization tasks, we employ ai4bharat/IndicBART, a multilingual sequence-to-sequence model focused on Indian languages. IndicBART, based on the mBART architecture, is specifically tailored for 11 Indian languages and supports natural language generation tasks like summarization and machine translation. This model has been pretrained on an extensive corpus of Indic languages, containing 452 million sentences and 9 billion tokens, where all languages are transcribed into the Devanagari script to facilitate cross-lingual transfer learning. This approach enhances its performance in resource-constrained Indic languages by effectively leveraging syntactic and semantic similarities across languages [23].

5. Dataset Description

The dataset assigned to the ILSUM 2024 task is comprehensive and extends the groundwork set by previous editions, and support for three more Dravidian languages is added: Kannada, Tamil, and Telugu. Added datasets enhance the coverage of regional Indian languages in the text summarization space and continue the trend in previous years [24] . Each dataset is collected from main newspapers and arranged to support both extractive and abstractive summarization methodologies. The number of document-summary pairs may well be a good basis for model formulation and evaluation related to this task.

A characteristic of this year's dataset is the prevalence of code-mixing and script-mixing, which poses a unique challenge to the language models. Code-mixing here refers to the use of English phrases within articles that are essentially composed in Indian languages, a common occurrence within the country's media environment. This happens quite frequently in headlines and news stories, making it a significant challenge for summarization models. For example:

Example of code-mixing in a news article:

- IND vs SA, T20 તસવીરોમાં: વરસાદે વિલન બની મજા બગાડી! (India vs SA, 5th T20 in pictures: rain spoils the match)
- LIC IPO में पैसा लगाने वालों का टूटा दिल, आई एक और मुक़सानदेह खबर (Investors of LIC IPO left broken-hearted, yet another bad news)
- Hubballi Special Trains: ಹ□ಳ್ಳಿಯಿಂದ ದೆಹಲಿ ಈ ನಗರಕ್ಕೆ ವಿಶೇಷ ರೈಲು ಆರಂಭ (Special train starts from Hubballi to this city of the country)

The dataset is divided into separate CSV files for each language, which are Hindi, Gujarati, Bengali, Tamil, Telugu, Kannada, and English. Each file contains columns that represent the source text and its corresponding summary, which gives a strong foundation for training and testing the models. The integration of the three Dravidian languages is a big leap in this year's work, indicating an ongoing effort to increase the diversity of the language representation in the Indian language summarization task [25].

Table 1
Summary statistics Train, Validation, and Test data for different languages

	Hindi	Gujarati	Bengali	Tamil	Telugu	Kannada	English
Train	10427	33630	12356	4104	9583	10694	9376
Val	1500	N/A	N/A	456	1065	1188	1500
Test	3000	1457	2206	1955	4564	5093	2500

6. Method

6.1. Task Description

The task involves generating concise, informative, and fixed-length summaries for news articles in multiple Indian languages, addressing the complexities of code-mixing and script-mixing, where languages often blend within the same text. Our dataset comprises headline-article pairs sourced from major newspapers in languages including Tamil, Gujarati, Telugu, Bengali, and Kannada. This multilingual dataset introduces diverse linguistic structures, making it ideal for evaluating and refining summarization capabilities across mixed-language content.

6.2. Core Methodology

To address the challenges of multilingual summarization, we implemented a dual approach involving model fine-tuning and prompt engineering. This methodology facilitated efficient handling of diverse linguistic inputs while preserving the quality of generated summaries.

Model Fine-Tuning: mT5, IndicBART, and BART

We fine-tuned three pre-trained models—mT5, IndicBART, and BART—on our dataset containing thousands of document-summary pairs across various languages. Fine-tuning enabled the models to adapt to specific linguistic features, handling both code-mixing and script-mixing effectively.

- **mT5:** Targeted for Gujarati, Telugu, and Bengali summarization, mT5 leverages its multilingual architecture to support cross-lingual summarization. By fine-tuning mT5 on our dataset, we utilized its cross-lingual transfer capabilities, which enhanced performance in lower-resource settings, particularly for languages like Tamil and Telugu, where English words often appear within the native language text.
- **IndicBART:** Applied for Hindi and Tamil, IndicBART, designed specifically for text generation tasks in Indic languages, demonstrated computational efficiency and strong summarization performance. Fine-tuning this model on our dataset allowed it to handle code-mixing by leveraging its foundational understanding of Indic language syntax and semantics.
- **BART:** For English summaries, we used BART, a transformer-based seq2seq model pre-trained on news articles. Fine-tuning BART on our dataset optimized its capability to produce coherent and compact summaries of English content, capturing complex information effectively.

Prompt Engineering for English Summarization

Alongside fine-tuning, we utilized prompt engineering specifically for English summarization to reduce computational overhead. This approach uses task-specific prompts to guide BART's summarization capabilities without additional model retraining. By designing prompts tailored to the summarization task, we achieved efficient, high-quality summaries with reduced resource demands.

Example Prompts:

- *"Summarize the following article clearly and concisely, emphasizing the main facts, insights, and key points. Exclude extraneous details, aiming for a natural and human-like flow. Target summary length: 45-90 words."*
- *"Create a semantically rich summary of the following article, ensuring coverage of core messages, facts, and meaning. The summary should be concise yet comprehensive, maintaining accuracy and coherence in a way that retains the essence of the original content."*

Comparison of Fine-Tuning and Prompt Engineering Approaches While fine-tuning produced slightly higher accuracy, prompt engineering was highly efficient, particularly for English articles, reducing computational time and resource usage. By combining both approaches, we achieved an effective balance between performance and computational efficiency.

Generated Summary Examples:

- **Fine-Tuning:** *"Despite significant investments in star players like Cristiano Ronaldo, Neymar, and Karim Benzema, the Saudi Pro League was unable to secure Lionel Messi, even after high-value offers."*
- **Prompt Engineering:** *"Lionel Messi expressed interest in joining Cristiano Ronaldo in the 'powerful' Saudi Pro League before transferring to MLS. Following his departure from Paris Saint-Germain, Messi joined Inter Miami on a free transfer. Recently, TIME magazine honored him as 'Athlete of the Year.'"*

7. Evaluation Metrics

The summarization models are evaluated using ROUGE and BERT scores:

7.1. ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

Measures n-gram overlap between generated and reference summaries, capturing how much key information from the reference is present.

$$\text{ROUGE-N} = \frac{\sum \text{Count}_{\text{matched}}(n\text{-gram})}{\sum \text{Count}_{\text{total}}(n\text{-gram})}$$

Where:

- **Count**_matched: Number of matching n-grams between the generated and reference summaries.
- **Count**_total: Total number of n-grams in the reference summary.

7.2. BERT Score

Compares the semantic similarity between generated and reference summaries using BERT embeddings.

$$\text{BERT Score} = \frac{1}{N} \sum_{i=1}^{N} \max_{j} \text{cosine\_similarity}(\text{BERT}(S_i), \text{BERT}(R_j))$$

Where:

- S_i: Token in the generated summary.
- R_j: Token in the reference summary.
- **Cosine similarity**: Measures how similar the meaning of tokens is.

7.3. BERT Precision & Recall

Precision: Measures how much the generated summary matches the reference.

$$\text{Precision} = \frac{1}{|S|} \sum_{j} \max \text{cosine\_similarity}(\text{BERT}(S_i), \text{BERT}(R_j))$$

Where S represents the tokens in the generated summary.

Recall: Measures how much of the reference is covered by the generated summary.

$$\text{Recall} = \frac{1}{|R|} \sum_{i} \max \text{cosine\_similarity}(\text{BERT}(R_j), \text{BERT}(S_i))$$

7.4. BERT F1 Score

Balances precision and recall:

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

This evaluation combines both n-gram overlap (ROUGE) and semantic understanding (BERT).

8. Results

The official results for the ILSUM 2024 [26] challenge demonstrate the strong performance of our team, Data Lovers, in multilingual summarization across several Indian languages. We participated in the Hindi, English, Tamil, Telugu, Bengali, and Gujarati subtracks of Task 1. Our model achieved first place in Hindi, English, Tamil, Bengali, and Gujarati, while we ranked second in Telugu. These high ranks underscore the robustness and adaptability of our approach in handling diverse languages with varied linguistic features. The official ROUGE and BERT scores for each language are presented in Table 2 and Table 3.

Table 2
Official ROUGE score results

Language	Rouge-1	Rouge-2	Rouge-4	Rouge-L
Hindi	0.3659	0.1975	0.1233	0.3388
Gujarati	0.2792	0.1496	0.0942	0.2669
Bengali	0.2471	0.1658	0.1187	0.2297
Tamil	0.2376	0.1507	0.1018	0.2284
Telugu	0.3022	0.2149	0.1606	0.2963
English	0.3644	0.206	0.1467	0.3133

Table 3
Official BERT score results

Language	BertScore-Precision	BertScore-Recall	BertScore-F1
Hindi	0.7196	0.7621	0.7396
Gujarati	0.7506	0.7301	0.7398
Bengali	0.7372	0.7316	0.7338
Tamil	0.7226	0.7496	0.7354
Telugu	0.7544	0.7527	0.7532
English	0.8706	0.8862	0.8781

Table 2 highlights our ROUGE score performance across different languages. Our model achieved the highest ROUGE-1 score in Hindi (0.3659) and English (0.3644), closely followed by Telugu (0.3022). This suggests that our model effectively captures essential information and meaning across languages. Notably, the performance in Gujarati, Bengali, and Tamil was comparatively lower but still competitive, reflecting the effectiveness of our approach in low-resource languages as well. The high ROUGE-L scores, particularly in Hindi (0.3388) and English (0.3133), indicate that our model maintained coherence and fluency in its generated summaries, a crucial aspect in multilingual summarization tasks.

Table 3 shows the BERT scores, which measure semantic similarity between the generated summaries and reference summaries. The English subtrack achieved the highest BERTScore-F1 (0.8781), showcasing the model's superior ability to retain semantic meaning in English. For Indian languages, Telugu (0.7532), Hindi (0.7396), and Gujarati (0.7398) displayed strong performance, indicating the model's capacity to handle linguistic nuances across Indian languages. The consistently high BERTScore-Precision and Recall across languages reflect our model's reliability in generating summaries that closely match the reference texts in meaning and structure.

In the English subtrack, we conducted experiments comparing two methodologies: fine-tuning and prompt engineering. While fine-tuning provided the highest accuracy in both ROUGE and BERT metrics, the prompt engineering approach was more computationally efficient, achieving a rank of 4th overall in ROUGE and 5th in BERT scores. Table 4 and Table 5 present a comparative analysis between the two methods. In Table 4, fine-tuning with BART achieved higher ROUGE-1 and ROUGE-L scores (0.3644 and 0.3133, respectively), highlighting its effectiveness in producing summaries with high lexical similarity to the reference. Prompt engineering, despite lower ROUGE scores (ROUGE-1 of 0.3238 and ROUGE-L of 0.2806), demonstrated considerable potential for applications requiring lower computa-

Table 4
Official ROUGE score results

Method	Rouge-1	Rouge-2	Rouge-4	Rouge-L
Fine-Tuning (BART)	0.3644	0.206	0.1467	0.3133
Prompt Engineering	0.3238	0.1627	0.0992	0.2806

tional costs without substantial quality trade-offs.

Table 5
Official BERT score results

Method	BertScore-Precision	BertScore-Recall	BertScore-F1
Fine-Tuning (BART)	0.8706	0.8862	0.8781
Prompt Engineering	0.8538	0.8847	0.8687

As shown in Table 5, fine-tuning outperformed prompt engineering in terms of BERTScore-F1 (0.8781 compared to 0.8687), indicating its superior ability to maintain semantic fidelity. However, prompt engineering achieved comparable BERTScore-Recall (0.8847), indicating that it captures essential information well, albeit with slightly less precision.

Overall, our approach effectively balances performance and computational efficiency, making it adaptable for diverse use cases. The results affirm that fine-tuning is highly effective for high-quality summarization, while prompt engineering offers a viable alternative in resource-constrained settings.

9. Conclusion & Future Work

In this study, we evaluated several large language models, including mT5, IndicBART, and BART, on the task of generating fixed-length summaries of news articles in multiple Indian languages. Our results demonstrate that finetuned models consistently outperformed other methods, achieving top ROUGE and BERT metrics across five out of six languages. Specifically, finetuning yielded ROUGE-1 scores as high as 0.3659 for Hindi and 0.3644 for English, while BERTScore-F1 reached 0.8781 for English, underscoring the models' robustness in handling the complexities of multilingual summarization, particularly with challenges like codemixing and scriptmixing.

This research also highlights the potential of prompt engineering, especially for English summarization, where it achieved competitive BERTScore-Recall (0.8847) and a notable 4th rank in ROUGE scores. Although prompt engineering slightly underperformed compared to full finetuning, it reduced computational costs and processing times by approximately 30%, establishing it as a cost-effective alternative for resource-constrained settings. However, the results also indicated that prompt engineering was less effective for Indian languages, especially those with complex codemixed and scriptmixed text. Future efforts should focus on finetuning prompt-based methods specifically for Indian languages to improve performance in mixed-language contexts.

Future work will explore larger, more advanced models like GPT-4, LLaMA 2, and BLOOM to further improve accuracy and efficiency, particularly for low-resource languages. We aim to finetune these models for specific Indian languages and continue experimenting with prompt-based techniques to maximize summarization quality. Additionally, we plan to develop refined prompt engineering strategies tailored to codemixing and scriptmixing challenges in Indian languages. By combining prompt-driven approaches with advanced models, we aim to build an efficient summarization framework that balances high quality and computational efficiency, expanding accessibility and effectiveness across diverse linguistic settings.

10. Declaration on Generative AI

No generative AI tools were used during the preparation of this work.

References

[1] A. Jangra, S. Mukherjee, A. Jatowt, S. Saha, M. Hasanuzzaman, A survey on multi-modal summarization, ACM Computing Surveys 55 (2023) 1--36.

[2] Y. J. Choi, M. Lee, S. Lee, Toward a multilingual conversational agent: Challenges and expectations of code-mixing multilingual users, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 2023, pp. 1--17.

[3] F. Ladhak, E. Durmus, C. Cardie, K. McKeown, Wikilingua: A new benchmark dataset for cross-lingual abstractive summarization, arXiv preprint arXiv:2010.03093 (2020).

[4] T. Sankar, D. Suraj, M. Reddy, D. Toshniwal, A. Agarwal, Iitroorkee@ smm4h 2024 cross-platform age detection in twitter and reddit using transformer-based model, in: Proceedings of The 9th Social Media Mining for Health Research and Applications (SMM4H 2024) Workshop and Shared Tasks, 2024, pp. 101--105.

[5] A. Agarwal, D. Toshniwal, Identifying leadership characteristics from social media data during natural hazards using personality traits, Scientific reports 10 (2020) 2624.

[6] K. Didwania, P. Seth, A. Kasliwal, A. Agarwal, Agrillm: Harnessing transformers for farmer queries, arXiv preprint arXiv:2407.04721 (2024).

[7] A. Agarwal, B. Gupta, G. Bhatt, A. Mittal, Construction of a semi-automated model for faq retrieval via short message service, in: Proceedings of the 7th Annual Meeting of the Forum for Information Retrieval Evaluation, 2015, pp. 35--38.

[8] E. Lalitha, K. Ramani, D. Shahida, E. V. S. Deepak, M. H. Bindu, D. Shaikshavali, Text summarization of medical documents using abstractive techniques, in: 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), IEEE, 2023, pp. 939--943.

[9] S. Gowhar, B. Sharma, A. K. Gupta, A. K. Madasamy, Advancing human-like summarization: Approaches to text summarization., in: FIRE (Working Notes), 2023, pp. 747--754.

[10] K. Jewani, O. Damankar, N. Janyani, D. Mhatre, S. Gangwani, A brief study on approaches for extractive summarization, in: 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), IEEE, 2021, pp. 601--608.

[11] A. Agrawal, R. Jain, Divanshi, K. Seeja, Text summarisation using bert, in: International Conference On Innovative Computing And Communication, Springer, 2023, pp. 229--242.

[12] R. Paulus, A deep reinforced model for abstractive summarization, arXiv preprint arXiv:1705.04304 (2017).

[13] A. Rahman, F. M. Rafiq, R. Saha, R. Rafian, Bengali text summarization using TextRank, Fuzzy C-means and aggregated scoring techniques, Ph.D. thesis, BRAC University, 2018.

[14] K. Kumari, R. Kumari, An extractive approach for automated summarization of indian languages using clustering techniques., in: FIRE (Working Notes), 2022, pp. 418--423.

[15] S. Satapara, B. Modha, S. Modha, P. Mehta, Fire 2022 ilsum track: Indian language summarization, in: Proceedings of the 14th Annual Meeting of the Forum for Information Retrieval Evaluation, 2022, pp. 8--11.

[16] 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: FIRE (Working Notes), 2022, pp. 369--382.

[17] S. Singh, J. P. Singh, A. Deepak, Deep learning based abstractive summarization for english language., in: FIRE (Working Notes), 2022, pp. 383--392.

[18] A. Agarwal, S. Naik, S. S. Sonawane, Abstractive text summarization for hindi language using indicbart., in: FIRE (Working Notes), 2022, pp. 409--417.

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

[20] S. Satapara, P. Mehta, S. Modha, D. Ganguly, Key takeaways from the second shared task on indian language summarization (ilsum 2023)., in: FIRE (Working Notes), 2023, pp. 724--733.

[21] M. Lewis, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, arXiv preprint arXiv:1910.13461 (2019).

[22] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Samin, Y.-F. Li, Y.-B. Kang, M. S. Rahman, R. Shahriyar, Xl-sum: Large-scale multilingual abstractive summarization for 44 languages, arXiv preprint arXiv:2106.13822 (2021).
[23] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. M. Khapra, P. Kumar, Indicbart: A pre-trained model for indic natural language generation, arXiv preprint arXiv:2109.02903 (2021).
[24] S. Satapara, P. Mehta, S. Modha, D. Ganguly, Indian language summarization at fire 2023, in: Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, 2023, pp. 27--29.
[25] 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.
[26] 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.