Text Summarization and Detection of Factual Incorrectness for Indian Languages

Durairaj Thenmozhi, Rohan R, Niranjana A, Madussree Ravi and Padmashri R

Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India

Abstract

As the volume of textual data in daily life continues to grow, developing strategies to produce concise and coherent summaries have become essential for facilitating efficient access to information. Text summarization involves condensing a large body of text while preserving essential information and key points. While numerous summarization techniques have been developed for resource-rich languages like English, there is a notable scarcity of methods tailored for Indian languages. This paper addresses text summarization tasks for news articles in Indian languages as part of the Indian Language Summarization (ILSUM) shared task of FIRE 2024. Task 1 is traditional text summarization, entailing generating abstractive summaries for news articles in English, Gujarati, Tamil and Telugu. We utilize the mT5_m2m_CrossSum model for cross-lingual abstractive summarization. Task 2 is detection of factual incorrectness in machine-generated summaries in Hindi and Gujarati. The classification of incorrectness types is done with the help of Logistic Regression. The performance of the chosen methodologies was evaluated using ROUGE metrics and BERT score for Task 1, while macro-F1 score was employed to evaluate Task 2. In the rank list released by the organizers, our team, Squad, achieved ranks of 8th, 5th, 6th, and 5th for the English, Gujarati, Tamil, and Telugu datasets respectively, in Task 1, and ranked 2nd for both Hindi and Gujarati datasets in Task 2.

Keywords

Text Summarization, Indian Languages, Abstractive Summarization, mT5_m2m_CrossSum, Factual Incorrectness, Logistic Regression, Multi-label Classification

1. Introduction

In today's data-driven world, the vast influx of textual information creates a challenge for individuals who have limited time to process and comprehend all the data available to them. Here, text summarization proves invaluable by condensing information, enabling quicker and easier consumption of essential content. Text summarization systems have diverse applications, such as generating concise overviews of web pages in search engines or distilling key information from lengthy articles to provide critical insights in a short time.

The concept of automatic text summarization originated in the 1950s with pioneering efforts at IBM Research Laboratories [1]. In recent years, advancements in the field have accelerated, largely due to the growth of the Internet, which has expanded both the volume of available data and the tools available for machines to generate summaries. Indian languages can be broadly divided into Indo-Aryan languages (e.g. Hindi-Urdu, Assamese, Bengali, Gujarati, Marathi) and Dravidian languages (e.g. Malayalam, Tamil, Telugu, Kannada) [2]. This paper aims to provide constructive insights into text summarization for various Indian languages, focusing on the accuracy and factual correctness of generated summaries.

Despite the prevalence of online summarizers, which many people now rely on, not all systems are capable of producing reliable and accurate summaries. Machine-generated summaries often face challenges such as misinterpretation of the text, factual inaccuracies, or "hallucinations", where the model generates entirely fabricated information that appears credible. These issues highlight the importance of verifying the factual correctness of machine-generated summaries, either by confirming their accuracy or identifying errors. This paper addresses the problem by taking the latter approach—detecting incorrectness within summaries.

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

theni_d@ssn.edu.in (D. Thenmozhi); rohan2210124@ssn.edu.in (R. R); niranjana2210379@ssn.edu.in (N. A); madussree2310250@ssn.edu.in (M. Ravi); padmashri2310123@ssn.edu.in (P. R)

The remainder of the paper is structured as follows: Section 2 reviews related work on text summarization for Indian languages; Section 3 describes the datasets used; Section 4 explains the methodology for the tasks; Section 5 outlines the performance metrics; Section 6 presents and analyzes the results; and Section 7 concludes the paper.

2. Related Works

This paper aims to expand on prior research in text summarization and factual misinformation detection, focusing on the linguistic nuances of Indian languages. The primary goal is to tailor summarization and misinformation detection techniques to address language-specific challenges.

In the realm of summarization, recent works [3] explores the use of mT5-small and mT5-base models, fine-tuning T5-base in both Indian English and Hindi. This approach merges the strengths of T5-base and translation models, which helps overcome linguistic challenges and enhances summary quality. Another study [4] employs extractive summarization using K-means clustering, with text tokenized and vectorized via Word2Vec, showing the adaptability of extractive techniques for both small and large data samples. Similarly, research [5] applies K-means clustering with Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to capture word frequency in sentences, further improving extractive summarization efficacy.

An article [6] provides an extensive review of summarization techniques for Indian languages, examining statistical, linguistic, machine learning, and hybrid methodologies. While machine learning models and neural networks analyze sentence relevance, dataset limitations affect their effectiveness. This study highlights the unique summarization challenges in Indian languages, offering evaluations for improved summarization approaches.

IndicBART, introduced in [7], is a multilingual sequence-to-sequence model designed to leverage similarities in Indic scripts, optimizing transfer learning for Indian languages. The paper demonstrates the competitive performance of IndicBART in Neural Machine Translation (NMT) compared to larger models like mBART50, particularly in low-resource environments.

Regarding misinformation detection, [8] investigates factual inaccuracies in machine-generated summaries, proposing a prompt-based classification system to identify errors such as misrepresentation and false attribution. While effective in leveraging transfer learning through natural language prompts, the study highlights the need for improvement in error detection across machine-generated summaries.

The Label Mask Multi-label Text Classification model (LM-MTC) presented in [9] uses masked language modeling to identify label correlations by assigning tokens to labels and masking them, drawing inspiration from cloze questions. This method enhances generalization, yielding strong results across datasets.

Studies [10] and [11] addressing classification challenges use up-sampling during preprocessing to counter class imbalance, significantly improving accuracy and identifying minority classes more effectively. By generating synthetic examples for underrepresented classes, these methods ensure balanced dataset learning.

An overview paper [12] examines multi-label text classification models within Indian languages like Hindi and Marathi, focusing on fine-tuned transformer and multilingual models. Despite notable F1-scores, the paper concludes that fine-grained multi-label classification remains challenging for language-specific datasets.

Lastly, insights from similar shared tasks [13] [14] [15] organized by FIRE provided valuable perspective, refining the development of our solution through comparative analysis with prior work. These tasks helped shape the methodological approach, enhancing the applicability and effectiveness of the solutions presented.

3. Dataset Description

The ILSUM 2024 shared task [16] [17] [18] [19] [20] addresses the gap in automatic summarization resources for Indian languages, where large datasets are scarce. This third edition expands coverage by introducing Kannada, Tamil, and Telugu, in addition to Hindi, Gujarati [21], Bengali, and Indian English, aiming to create reusable corpora for effective summarization.

3.1. Task 1

The given task focuses on traditional text summarization, with the goal of generating concise, meaningful fixed-length summaries for news articles available in multiple Indian languages. The datasets provided are in English, Gujarati, and Dravidian languages such as Tamil and Telugu. The dataset for each language is divided into three parts: training, validation, and test. The training and validation datasets each contain a unique identifier, the news article text, its headline, and the corresponding summary. The test dataset consists of id and headline-article pairs. Refer to Table 1 for detailed dataset statistics of each language.

Table 1
Dataset Statistics for Task 1

	English	**Gujarati**	**Tamil**	**Telugu**
Training Data	9376	12356	4104	9583
Validation Data	1500	–	456	1065
Test Data	2500	1457	1955	4564
Total	13376	13813	6515	15212

3.2. Task 2

This task aims to detect factual inaccuracies in machine-generated summaries of English news articles provided in Indian languages. It includes two subtasks based on the summary language: Hindi and Gujarati. The training dataset contains the source news title, headline, and article in English, along with both factually correct and incorrect abstractive summaries in English, Hindi, and Gujarati, as well as the type of incorrectness. The various categories of summaries are:

- Misrepresentation: It involves conveying information in a misleading way, often by exaggerating, downplaying, or distorting facts to fit a specific narrative and is a type of factual incorrect summary.
- Inaccurate quantities: It is the factual incorrectness that arises when specific quantities, measurements, or statistics are misrepresented, intentionally or accidentally.
- False Attribution: It is another form of factual incorrectness, involving misattributing a statement, idea, or action to an individual or a group.
- Fabrication: It is a serious form of factual incorrectness, involving the creation of entirely false data, sources, or events.
- Correct: It refers to the factually correct summary of the given article.

The train dataset consists of 4975 records; Table 2 describes the distribution of the same, before and after upsampling. The test dataset for both Hindi and Gujarati consists of 200 records each; comprising the id, source English article, and corresponding summary in the respective Indian language.

Table 2
Dataset Distribution for Task 2

Labels	Before Upsampling	After Upsampling
Misrepresentation	294	3986
Incorrect quantities	195	3986
False attribution	250	3986
Fabrication	250	3986
Correct	3986	3986
Total	4975	19930

4. Methodology

4.1. Task 1

4.1.1. Text Preprocessing

The methodology begins with preprocessing the input articles to ensure a clean and standardized text format. All articles undergo a cleaning process where unnecessary whitespace and line breaks are removed. This step is essential for providing a consistent input text that maximizes model efficiency and minimizes noise, setting a solid foundation for subsequent summarization tasks.

4.1.2. Model Selection and Summary Generation

For summary generation, we experimented with four models—BART, T5, mT5, and mT5_m2m_CrossSum—and selected the mT5_m2m_CrossSum model [22] as it delivered the best results. This model, a variant of the mT5 (Multilingual T5) architecture that incorporates features from the m2m model, is pre-trained and fine-tuned on CrossSum, the largest cross-lingual summarization dataset, which supports effective summarization across numerous language pairs [23]. The mT5_m2m_CrossSum model is particularly well-suited for this task, as it is optimized for multilingual contexts and can generate high-quality summaries across diverse languages.

In implementing this model, we used specific hyperparameters to enhance performance and quality. The input text was tokenized with a maximum length of 512 tokens, with summaries generated up to a maximum of 84 tokens. To improve coherence and prevent repetition, we set no_repeat_ngram_size to 2. Beam search with num_beams set to 4 was used to maintain output fluency. Additionally, the decoder_start_token_id was set to the ID corresponding to the target language to facilitate language-specific summarization. By leveraging these configurations, we achieved seamless cross-lingual summarization, enabling accurate summaries across multiple Indian languages, aligned with the multilingual nature of the input dataset.

4.1.3. Tokenization and Language Specification

Following text preprocessing, the cleaned articles are tokenized using the mT5 tokenizer, a component specifically designed to handle multilingual input effectively. Tokenization is an essential step, as it converts the raw text into a format that the model can process. During the tokenization process, the target language of the summary is specified. This enables the transformer model to decode the text in the desired language accurately, which is crucial for multilingual datasets. The mT5 tokenizer provides robust tokenization tailored to the characteristics of each language in the dataset, facilitating effective input handling and enhancing the model's performance.

4.1.4. Evaluation with ROUGE Metrics

The generated summaries are evaluated using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics, a widely used metric for summarization tasks. ROUGE measures the n-gram, word

sequence, and word pair overlap between the generated summaries and reference summaries, providing a quantitative assessment of summary quality. We analyze ROUGE scores for each language, focusing on ROUGE-1, ROUGE-2, and ROUGE-L metrics, to capture different aspects of summarization accuracy. This evaluation approach allows for a nuanced understanding of the model's performance across languages, assessing both content relevance and structure fidelity.

This methodology combines rigorous preprocessing, advanced multilingual modeling, and robust evaluation metrics to provide high-quality cross-lingual summarization. By standardizing input text, leveraging the specialized mT5_m2m_crossSum model, and analyzing summary accuracy with ROUGE metrics, this approach ensures effective and scalable summarization across multiple Indian languages, making it a valuable asset for multilingual content summarization.

4.2. Task 2

4.2.1. Data Cleaning

The methodology begins by addressing missing values in the Incorrectness_Type column, where null entries are replaced with the label "Correct". This approach allows for consistency in the dataset by ensuring that all records have a defined label, facilitating smoother processing and analysis. To further normalize the dataset, summaries corresponding to these entries are copied into the Incorrectness_Type column, establishing a uniform structure.

4.2.2. Upsampling the Data

Recognizing the challenge of class imbalance in the dataset, an upsampling technique is applied to equalize the representation of each class. This ensures that the model is not biased towards more frequent labels, improving its ability to generalize across all categories of incorrectness.

4.2.3. Feature Extraction

For feature extraction, the Article, Incorrect_Summary_Gujarati and Incorrect_Summary_Hindi columns are combined into a single textual input feature. By merging these fields, the model can access richer contextual information from both the original article and the provided summaries, potentially enhancing its understanding of correctness nuances. The combined text is then processed using a TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer, which converts textual data into a weighted numerical representation based on term relevance. This transformation captures the importance of specific terms across documents, enabling the model to focus on meaningful features.

4.2.4. Model Building

After vectorization, the dataset is divided into training and testing subsets, where the vectorizer is fit to the training data and subsequently applied to the test data to ensure consistency during training the model. A variety of traditional models, including Support Vector Machine, Logistic Regression, and Random Forest, as well as ensemble methods like Bagging Classifier and Gradient Boosting, were evaluated. Ultimately, Logistic Regression was identified as the top-performing model. A Logistic Regression model was then trained on the vectorized data, with a maximum of 1000 iterations, utilizing the transformed text to predict the Incorrectness_Type. This method is chosen for its efficiency and interpretability in multiclass classification tasks.

Overall, this approach maintains data integrity, mitigates class imbalance, and uses text vectorization to enhance predictive accuracy, offering a structured and scalable solution to identifying incorrectness types.

5. Performance Metrics

This section provides insight on the metrics used to evaluate the performance of the methodologies employed for each task.

5.1. Task 1

5.1.1. ROUGE Score

The summaries generated for each language are evaluated using the standard ROUGE (Recall-Oriented Understudy for Gisting Evaluation), which indicates the extent of similarity between the reference human-created summary and the machine-generated summary.

- **ROUGE-N:** Measures the overlap of n-grams, which are contiguous sequences of n items, between the generated summary and the reference summaries.
  - ROUGE-1: Considers unigrams (single words).
  - ROUGE-2: Considers bigrams (pairs of consecutive words).
  - ROUGE-4: Considers sequences of 4 words.
- **ROUGE-L:** Measures the longest common subsequence between the generated and reference summaries, taking into account the order of words and enhancing sensitivity to the summary's structure.

The ROUGE scores are calculated by considering the Recall, Precision, and F1 scores.

- **Recall:** Refers to the extent to which the generated summary aligns with the reference summary, specifically indicating the proportion of sentences selected by the human that were accurately recognized by the system.

$$\text{Recall} = \frac{\text{Number of overlapping n-grams}}{\text{Total n-grams in reference}} \tag{1}$$

- **Precision:** Indicates the proportion of the generated summary that is relevant or necessary; it represents the fraction of sentences produced by the system that are accurate.

$$\text{Precision} = \frac{\text{Number of overlapping n-grams}}{\text{Total n-grams in generated summary}} \tag{2}$$

- **F1 Score:** The harmonic mean of precision and recall, providing a balance between the two.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

5.1.2. BERT Score

The evaluation of the quality of text summarization—measuring how similar the text summary is to the original text—can be done using the BERT (Bidirectional Encoder Representations from Transformers) Score metric. It leverages contextual embeddings from the BERT model to capture the semantic similarity between the generated text and reference text summaries. The generated and reference texts are tokenized and mapped to their corresponding BERT embeddings, which are compared using cosine similarity.

- **Recall:** Evaluates the proportion of relevant tokens in the generated summary that match the reference summary.
- **Precision:** Assesses how many tokens from the reference summary are effectively captured in the generated text.
- **F1 Score:** The harmonic mean of precision and recall, providing a balanced score.

ROUGE is more appropriate for extractive summarization, as it relies on exact n-gram overlap, which aligns with the goals of extractive methods. BERT Score is better suited for abstractive summarization because it captures meaning beyond structural similarity, evaluating summaries based on semantic similarity.

5.2. Task 2

For the evaluation of our methodology for Task 2 , we have chosen Precision, Recall and F1-Score as performance metrics.

- Recall for a specific label indicates the proportion of correctly identified instances of that label out of all true instances in the data.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4}$$

- Precision for a specific label (incorrectness type) measures the proportion of correctly identified instances of that type out of all instances predicted as that type by the model.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{5}$$

- F1 Score for a specific label is the harmonic mean of precision and recall for that label, providing a balanced measure.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

The Macro-Average F1 Score evaluates the model's performance across all classes in multi-label classification by calculating the F1 score for each class independently, ensuring that each class is given equal weight in the overall score.

To set a wider table, which takes up the whole width of the page's live area, use the environment `table*` to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table **??** is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed output of this document.

6. Result Analysis

6.1. Task 1

We experimented with four different text summarization models - BART, T5, mT5, and mT5_m2m_CrossSum — aiming to determine the most effective model for generating summaries of articles in English, Hindi, Tamil, and Telugu. The mT5_m2m_CrossSum model consistently outperformed the others, owing to its strong cross-lingual transfer capabilities and its tailored design for abstractive summarization across various languages. Using this model, we generated summaries for each test dataset, calculating ROUGE and BERT scores for each language, with the final performance results detailed in Table 3. For Task 1, our submissions achieved ranks of 8th for English, 5th for Gujarati, 6th for Tamil, and 5th for Telugu in the official rank list released by the organizers.

The cross-lingual architecture and pre-trained nature of the mT5_m2m_CrossSum model enhance its ability to handle linguistic nuances, resulting in high-quality, abstractive summaries. Further improvements in text summarization with this model are feasible by fine-tuning on domain-specific datasets, which would deepen contextual understanding and yield more cohesive summaries.

Table 3
Task 1 Performance Scores

	BERT Score			ROUGE Score			
Language	**Precision**	**Recall**	**F1 Score**	**ROUGE-1**	**ROUGE-2**	**ROUGE-4**	**ROUGE-L**
English	0.8764	0.8447	0.8601	0.2214	0.0579	0.0111	0.1761
Gujarati	0.7578	0.6844	0.7186	0.1810	0.0811	0.0347	0.1748
Tamil	0.6192	0.5747	0.5951	0.0121	0.0007	0.0001	0.0120
Telugu	0.7392	0.6765	0.7058	0.1498	0.0695	0.0226	0.1434

6.2. Task 2

After having experimented with various models such as Support Vector Machine, Random Forest, Bagging Classifier and Gradient Boosting, the Logistic Regression model was selected owing to its probabilistic nature that allows it to accurately distinguish among multiple classes based on the learned features. The performance results for each test dataset are summarized in Table 4. For Task 2, our submissions for both the Hindi and Gujarati datasets have secured 2nd rank in the rank list released by the organizers.

Table 4
Task 2 Performance Scores

Language	**Macro Avg F1 Score**
Hindi	0.3153
Gujarati	0.2960

The confusion matrix is shown in Figure 1 to depict the model's classification accuracy and highlight the distribution of predicted labels compared to the actual labels. The model demonstrated high accuracy for labels such as "fabrication," "incorrect quantities," and "misrepresentation," with no misclassifications in these categories. However, there were slight errors in differentiating between the "Correct" and "false attribution" labels, indicating that the features used to distinguish these classes may overlap and need further refinement to enhance classification accuracy.

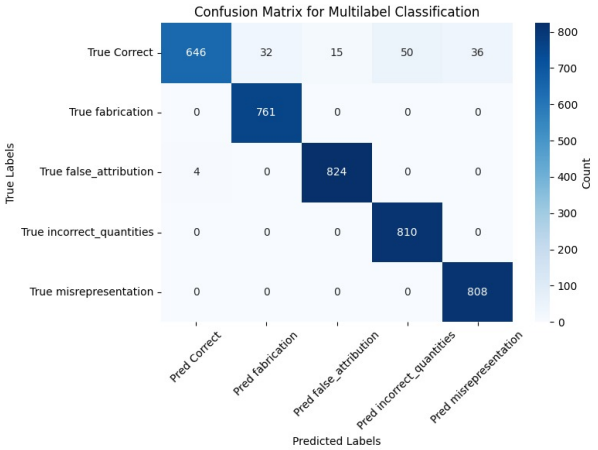

Figure 1: Confusion Matrix

Future work should focus on enhancing the Logistic Regression model for detecting factual incorrectness by implementing advanced feature engineering techniques, such as integrating semantic embeddings and linguistic attributes. Additionally, exploring advanced architectures like neural networks and large language models (LLMs), alongside ensemble methods and hyperparameter tuning,

could significantly improve overall model performance.

7. Conclusion

In conclusion, this paper has investigated text summarization tasks for Indian languages, focusing on two key areas. In Task 1, we employed the pre-trained model mT5_m2m_CrossSum for summarizing articles in languages such as English, Gujarati, Tamil, and Telugu. The model achieved favorable ROUGE scores, demonstrating a significant lexical overlap between the generated summaries and the reference summaries. Task 2 involved the detection of factually incorrect cross-lingual summaries in Hindi and Gujarati using a Logistic Regression model, which effectively classified the summaries into their respective categories. The evaluation and analysis of the results from both tasks provided valuable insights into the challenges of text summarization and classification in multilingual contexts. Ongoing advancements in model refinement and feature extraction will be crucial for enhancing performance in future research efforts.

Acknowledgments

We sincerely thank the organizers of the FIRE 2024 Indian Language Summarization (ILSUM) shared task for giving us the opportunity to participate and contribute to advancements in multilingual summarization and factual correctness detection. We are also deeply grateful to our institution, Sri Sivasubramaniya Nadar College of Engineering, and to our mentors for their unwavering support and guidance throughout the course of this research.

Declaration on Generative AI

The authors employed tools such as ChatGPT and Grammarly to assist with grammar and spelling correction, paraphrasing and rewording, during the preparation of this work. All content was subsequently reviewed and edited by the authors as needed and the authors take full responsibility for the publication's content.

References

[1] H. P. Luhn, The automatic creation of literature abstracts, IBM Journal of research and development 2 (1958) 159–165.

[2] P. Dhanya, M. Jathavedan, Comparative study of text summarization in indian languages, International Journal of Computer Applications 75 (2013).

[3] V. Ilanchezhiyan, R. Darshan, E. M. Dhitshithaa, B. Bharathi, Text summarization for indian languages: Finetuned transformer model application., in: FIRE (Working Notes), 2023, pp. 766–774.

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

[5] R. Khan, Y. Qian, S. Naeem, Extractive based text summarization using kmeans and tf-idf, International Journal of Information Engineering and Electronic Business 11 (2019) 33–44. doi:10.5815/ijieeb.2019.03.05.

[6] K. K. Mamidala, S. K. Sanampudi, Text summarization for indian languages: a survey, Int J Adv Res Eng Technol (IJARET) 12 (2021) 530–538.

[7] 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).

[8] A. Deroy, S. Maity, S. Ghosh, Prompted zero-shot multi-label classification of factual incorrectness in machine-generated summaries., in: FIRE (Working Notes), 2023, pp. 734–746.

[9] R. Song, Z. Liu, X. Chen, H. An, Z. Zhang, X. Wang, H. Xu, Label prompt for multi-label text classification, Applied Intelligence 53 (2023) 8761–8775.
[10] L. Muflikhah, A. Iskandar, N. Yudistira, B. N. Dewanto, I. U. Nadhori, L. K. Nisa, Up sampling data in bagging tree classification and regression decision tree method for dengue shock syndrome detection, in: Asia Simulation Conference, Springer, 2023, pp. 307–318.
[11] M. A. Hama Saeed, Diabetes type 2 classification using machine learning algorithms with up-sampling technique, Journal of Electrical Systems and Information Technology 10 (2023) 8.
[12] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schäfer, T. Ranasinghe, M. Zampieri, D. Nandini, et al., Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages, arXiv preprint arXiv:2112.09301 (2021).
[13] S. Singh, J. P. Singh, A. Deepak, Deep learning based abstractive summarization for english language., in: FIRE (Working Notes), 2022, pp. 383–392.
[14] M. Amjad, S. Butt, H. I. Amjad, A. Zhila, G. Sidorov, A. Gelbukh, Overview of the shared task on fake news detection in urdu at fire 2021, arXiv preprint arXiv:2207.05133 (2022).
[15] 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.
[16] 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.
[17] 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.
[18] S. Satapara, P. Mehta, D. Ganguly, S. Modha, Fighting fire with fire: Adversarial prompting to generate a misinformation detection dataset, arXiv preprint arXiv:2401.04481 (2024).
[19] 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.
[20] 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.
[21] S. Satapara, P. Mehta, S. Modha, D. Ganguly, Key takeaways from the second shared task on indian language summarization (ilsum 2023)., Working Notes of FIRE (2023) 724–733.
[22] T. Hasan, A. Bhattacharjee, W. U. Ahmad, Y.-F. Li, Y. bin Kang, R. Shahriyar, Crosssum: Beyond english-centric cross-lingual abstractive text summarization for 1500+ language pairs, CoRR abs/2112.08804 (2021). URL: https://arxiv.org/abs/2112.08804. arXiv:2112.08804.
[23] L. Xue, mt5: A massively multilingual pre-trained text-to-text transformer, arXiv preprint arXiv:2010.11934 (2020).