

CrossFactual: A Novel Approach for Detecting Factual Inaccuracies in Machine-Generated Summaries

Aniket Deroy^{1,*†}, Subhankar Maity¹

¹IIT Kharagpur, Kharagpur, India

Abstract

Detecting factual inaccuracy in machine-generated summaries is a novel and challenging task. Participants are tasked with identifying factual errors in summaries produced from English source documents, which are provided in Hindi and Gujarati. The training set includes English source documents along with summaries in English, Hindi, and Gujarati, enabling participants to familiarize themselves with error detection across languages. The test set consists solely of the English source document paired with summaries in Hindi and Gujarati. We focus on categorizing each data point based on the presence of factual inaccuracies, exploring four distinct types of factual errors. This study aims to enhance understanding of cross-lingual summary accuracy and contribute to improved evaluation frameworks in multilingual contexts. We use GPT-3.5 Turbo via prompting combined with several algorithmic approaches to detect factual inaccuracies in the machine-generated summaries across both languages. This paper presents a comparative analysis of factual inaccuracy detection models in Gujarati and Hindi, focusing on their performance across multiple experimental runs. The study reveals that Run 5 is the most effective model for both languages, achieving a F1 score of 0.0677, while other runs exhibit significantly lower scores, particularly Run 4. Notably, the ensemble approach demonstrates the highest performance results. Despite these advancements, the overall scores indicate ongoing challenges in creating robust models for detecting factual inaccuracies in Gujarati and Hindi. The findings emphasize the need for continued research and refinement to enhance the effectiveness of detection systems in these linguistic contexts.

Keywords

GPT, Factual Inaccuracies, Prompt Engineering, Hindi, Gujarati

1. Introduction

Detecting factual inaccuracy in machine-generated summaries presents a novel and challenging task, particularly in a multilingual context [1, 2]. As automated summarization technologies advance, ensuring the reliability of generated content becomes increasingly critical, especially when the output is intended for diverse language speakers [3]. This study focuses on identifying factual errors in summaries produced from English source documents, specifically targeting Hindi and Gujarati languages.

Participants are engaged in a rigorous evaluation process where they must identify inaccuracies within these summaries. To facilitate this, the training set comprises English source documents along with their corresponding summaries in English, Hindi, and Gujarati, allowing participants to develop a nuanced understanding of factual error detection across languages. The test set narrows this focus, providing only the English source document alongside summaries in Hindi and Gujarati, which encourages participants to apply their learned skills in a practical setting.

We emphasize the categorization of each data point based on the presence of factual inaccuracies, exploring four distinct types of factual errors. By examining these variations, we aim to provide insights into the nature of inaccuracies that can arise in machine-generated summaries. Additionally, we leverage the capabilities of GPT-3.5 Turbo [4], employing prompting techniques in conjunction with various algorithmic approaches to enhance the detection of factual discrepancies across both languages. This study ultimately aims to deepen our understanding of cross-lingual summary accuracy and contribute to the development of robust evaluation frameworks in multilingual contexts.

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

*Corresponding author.

✉ roydanik18@kgpian.iitkgp.ac.in (A. Deroy); subhankar.ai@kgpian.iitkgp.ac.in (S. Maity)

id 0000-0001-7190-5040 (A. Deroy); 0009-0001-1358-9534 (S. Maity)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

This paper offers a comparative analysis of models for detecting factual inaccuracies in Gujarati and Hindi, examining their performance across various experimental runs. The results indicate that Run 5 is the most effective model for both languages, achieving a F1 score of 0.0677, while other runs, particularly Run 4, show significantly lower scores. The ensemble approach stands out with the highest performance metrics. However, despite these improvements, the overall results highlight persistent challenges in developing robust detection models for factual inaccuracies in both languages. These findings underscore the necessity for ongoing research and refinement to improve the effectiveness of detection systems within these linguistic contexts.

2. Related Work

The task of detecting factual inaccuracies in machine-generated summaries has gained significant attention in recent years, driven by advancements in natural language processing (NLP) and the increasing reliance on automated summarization tools [5, 6]. A considerable body of work has focused on evaluating the quality of machine-generated text, particularly in terms of factual correctness and coherence [6, 7].

One of the early approaches in this domain involved manual evaluation of summaries, where human annotators assessed the fidelity of the content against the source material [8]. Studies by [9, 10] highlighted the importance of ensuring that summaries accurately represent the source, laying the groundwork for subsequent automated methods.

With the advent of deep learning, researchers began exploring automatic evaluation metrics for summarization. The introduction of ROUGE (Recall-Oriented Understudy for Gisting Evaluation) by [11] provided a quantitative method for assessing summary quality, although it primarily focuses on lexical similarity rather than factual correctness. To address this gap, recent studies have proposed metrics that consider factual consistency, such as FactCC and QAGS, which evaluate whether the generated summary maintains the truthfulness of the original content [12, 13].

In the realm of cross-lingual summarization, researchers like [14] and [15] have explored methods for generating and evaluating summaries across different languages. These studies emphasize the importance of understanding linguistic nuances and maintaining factual integrity when summarizing content in languages with distinct grammatical and syntactic structures. Furthermore, work by [16] highlights the challenges in cross-lingual settings, particularly when dealing with low-resource languages, and emphasizes the need for tailored evaluation frameworks.

Our approach builds upon this foundational work, particularly by incorporating both algorithmic and human-driven methods for detecting factual inaccuracies in multilingual contexts. Leveraging the capabilities of GPT-3.5 Turbo allows us to explore advanced prompting techniques that enhance accuracy detection, aligning with trends in using transformer-based models for nuanced language understanding [17]. This study aims to bridge the gap between existing methodologies and the specific challenges of cross-lingual factual accuracy, contributing valuable insights to the ongoing discourse in this evolving field.

3. Dataset

There are 200 (article,summary) pairs in Gujarati Language and 200 (article,summary) pairs in Hindi language in the test set respectively.

4. Task Definition

The task [18, 19, 20, 21, 22, 23, 24] is, given a Gujarati and Hindi summaries we have to classify the summaries into one of the five categories namely-Misrepresentation, False Attribution, Incorrect quantities, Fabrication and Correct.

5. Methodology

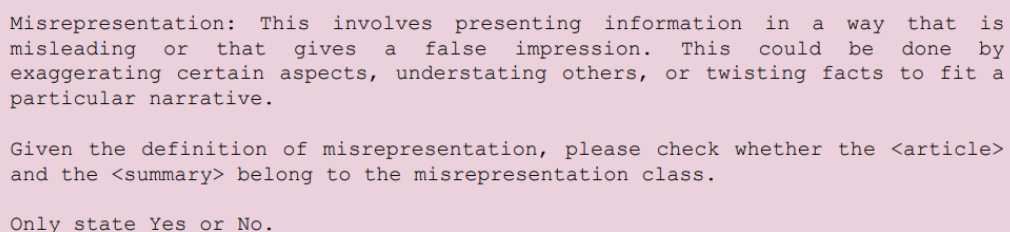
Prompting [25, 26] is a powerful technique that leverages large language models (LLMs) like GPT-3.5 Turbo to generate contextually relevant and accurate responses based on specific inputs. Here are several reasons why prompting is beneficial, particularly in the context of detecting factual inaccuracies in machine-generated summaries:

- **Flexibility and Adaptability:** Prompting allows researchers to customize the input to the model, guiding it to focus on specific tasks such as factual accuracy detection [27]. This adaptability enables a tailored approach that can be adjusted based on the nuances of the task or the languages involved.
- **Enhanced Contextual Understanding:** LLMs excel at understanding context due to their training on vast amounts of text [28]. By crafting well-designed prompts, we can help the model better grasp the relationships between the source document and the generated summary, facilitating more accurate assessments of factual correctness.
- **Efficiency in Error Detection:** Prompting can streamline the process of identifying factual inaccuracies by generating direct queries related to specific claims or statements in the summaries [29]. This efficiency reduces the need for extensive manual evaluation and allows for rapid analysis of multiple summaries.
- **Leveraging Knowledge:** LLMs possess a wealth of general knowledge and can often identify inaccuracies based on their understanding of facts and relationships [30]. By employing prompting, we can harness this knowledge to flag discrepancies in the summaries, even when they are not explicitly stated in the source material.
- **Multilingual Capabilities:** Given the cross-lingual nature of this study, prompting can be particularly advantageous in handling different languages [31]. The model's ability to process and generate text in multiple languages enhances its utility in evaluating summaries produced in Hindi and Gujarati from English sources.
- **Combining with Algorithmic Approaches:** Prompting can complement traditional algorithmic methods, creating a hybrid approach that combines the strengths of both [26]. This synergy can lead to more robust and comprehensive evaluations of factual accuracy.
- **Facilitating User Interaction:** Involving participants in the evaluation process through prompting can lead to more engaging interactions, as users can pose questions or seek clarifications, enhancing the overall assessment of factual accuracy [32].

Overall, prompting serves as a versatile tool that enhances the capabilities of LLMs in detecting factual inaccuracies, making it an integral part of our approach in this study.

5.1. Prompt Engineering-Based Approach combined with Algorithms

- For the Misrepresentation class the prompt is shown in Fig. 1:



```
Misrepresentation: This involves presenting information in a way that is misleading or that gives a false impression. This could be done by exaggerating certain aspects, understating others, or twisting facts to fit a particular narrative.

Given the definition of misrepresentation, please check whether the <article> and the <summary> belong to the misrepresentation class.

Only state Yes or No.
```

Figure 1: Prompt for the Misrepresentation class.

- For the Incorrect_Quantities class the prompt is depicted in Fig. 2:

```
Incorrect_Quantities: Factual incorrectness can occur when precise quantities,
measurements, or statistics are misrepresented, whether through error or
intent.

Given the definition of incorrect_quantities, please check whether the
<article> and the <summary> belong to the incorrect_quantities class.

Only state Yes or No.
```

Figure 2: Prompt for the Incorrect_Quantities class.

```
False_Attribution: Incorrectly attributing a statement, idea, or action to a
person or group is another form of factual incorrectness.

Given the definition of false_attribution, please check whether the <article>
and the <summary> belong to the false_attribution class.

Only state Yes or No.
```

Figure 3: Prompt for the False_Attribution class.

```
Fabrication: Making up data, sources, or events is a severe form of factual
incorrectness. This involves creating "facts" that have no basis in reality.

Given the definition of fabrication, please check whether the <article> and
the <summary> belong to the fabrication class.

Only state Yes or No.
```

Figure 4: Prompt for the Fabrication class.

- For the False_Attribution class the prompt is given in Fig. 3:
- For the Fabrication class the prompt is illustrated in Fig. 4:

We use the GPT-3.5 Turbo model at different temperature values via zero-shot prompting.

Next, we discuss four algorithms named Algorithm 1, Algorithm 2, Algorithm 3, and Algorithm 4.

The fifth approach is an ensembling approach where we run every algorithm(Algorithm 1-4) in three different temperature values 0.7, 0.8, and 0.9. Then we take an ensemble of all the runs by considering majority voting in which the label which occurs maximum no of times for a datapoint is selected. We perform the same for all the datapoints.

Next, we discuss the four algorithms namely Algorithm 1, Algorithm 2, Algorithm 3, and Algorithm 4.

Algorithm 1:

1. **Input:** A pair consisting of an article and its corresponding incorrect summaries (in Hindi or Gujarati).
2. **Step 1:**
 - Prompt the Large Language Model (LLM) with the pair (article and incorrect summaries) to determine if it belongs to the **Misrepresentation** class.
 - **If** the predicted label is **Misrepresentation**:
 - **Output:** Misrepresentation
 - **End** the algorithm for this datapoint.
3. **Step 2:**

- If the pair does not belong to the **Misrepresentation** class, prompt the LLM to check if it belongs to the **Fabrication** class.
- If the predicted label is **Fabrication**:
 - **Output**: Fabrication
 - **End** the algorithm for this datapoint.

4. **Step 3:**

- If the pair does not belong to the **Fabrication** class, prompt the LLM to check if it belongs to the **False Attribution** class.
- If the predicted label is **False Attribution**:
 - **Output**: False Attribution
 - **End** the algorithm for this datapoint.

5. **Step 4:**

- If the pair does not belong to the **False Attribution** class, prompt the LLM to check if it belongs to the **Incorrect Quantities** class.
- If the predicted label is **Incorrect Quantities**:
 - **Output**: Incorrect Quantities
 - **End** the algorithm for this datapoint.

6. **Step 5:**

- If the pair does not belong to any of the above classes, classify it as **Correct**.

7. **Repeat** this procedure for every datapoint in the dataset.

Algorithm 2:

1. **Input**: A pair consisting of an article and its corresponding incorrect summaries (in Hindi or Gujarati).

2. **Step 1:**

- Prompt the Large Language Model (LLM) with the pair (article and incorrect summaries) to determine if it belongs to the **Fabrication** class.
- If the predicted label is **Fabrication**:
 - **Output**: Fabrication
 - **End** the algorithm for this datapoint.

3. **Step 2:**

- If the pair does not belong to the **Fabrication** class, prompt the LLM to check if it belongs to the **Misrepresentation** class.
- If the predicted label is **Misrepresentation**:
 - **Output**: Misrepresentation
 - **End** the algorithm for this datapoint.

4. **Step 3:**

- If the pair does not belong to the **Misrepresentation** class, prompt the LLM to check if it belongs to the **False Attribution** class.
- If the predicted label is **False Attribution**:
 - **Output**: False Attribution
 - **End** the algorithm for this datapoint.

5. **Step 4:**

- If the pair does not belong to the **False Attribution** class, prompt the LLM to check if it belongs to the **Incorrect Quantities** class.
- If the predicted label is **Incorrect Quantities**:

- **Output:** Incorrect Quantities
 - **End** the algorithm for this datapoint.
6. **Step 5:**
- **If** the pair does not belong to any of the above classes, classify it as **Correct**.
7. **Repeat** this procedure for every datapoint in the dataset.

Algorithm 3:

1. **Input:** A pair consisting of an article and its corresponding incorrect summaries (in Gujarati or Hindi).
2. **Step 1:**
 - Prompt the Large Language Model (LLM) with the pair (article and incorrect summaries) to determine if it belongs to the **False_Attribution** class.
 - **If** the predicted label is **False_Attribution**:
 - **Output:** False_Attribution
 - **End** the algorithm for this datapoint.
3. **Step 2:**
 - **If** the pair does not belong to the **False_Attribution** class, prompt the LLM to check if it belongs to the **Misrepresentation** class.
 - **If** the predicted label is **Misrepresentation**:
 - **Output:** Misrepresentation
 - **End** the algorithm for this datapoint.
4. **Step 3:**
 - **If** the pair does not belong to the **Misrepresentation** class, prompt the LLM to check if it belongs to the **Fabrication** class.
 - **If** the predicted label is **Fabrication**:
 - **Output:** Fabrication
 - **End** the algorithm for this datapoint.
5. **Step 4:**
 - **If** the pair does not belong to the **Fabrication** class, prompt the LLM to check if it belongs to the **Incorrect Quantities** class.
 - **If** the predicted label is **Incorrect Quantities**:
 - **Output:** Incorrect Quantities
 - **End** the algorithm for this datapoint.
6. **Step 5:**
 - **If** the pair does not belong to any of the above classes, classify it as **Correct**.
7. **Repeat** this procedure for every datapoint in the dataset.

Algorithm 4:

1. **Input:** A pair consisting of an article and its corresponding incorrect summaries (in Gujarati or Hindi).
2. **Step 1:**
 - Prompt the Large Language Model (LLM) with the pair (article and incorrect summaries) to determine if it belongs to the **Incorrect_Quantities** class.
 - **If** the predicted label is **Incorrect_Quantities**:
 - **Output:** Incorrect_Quantities
 - **End** the algorithm for this datapoint.

3. **Step 2:**

- **If** the pair does not belong to the **Incorrect_Quantities** class, prompt the LLM to check if it belongs to the **Misrepresentation** class.
- **If** the predicted label is **Misrepresentation**:
 - **Output**: Misrepresentation
 - **End** the algorithm for this datapoint.

4. **Step 3:**

- **If** the pair does not belong to the **Misrepresentation** class, prompt the LLM to check if it belongs to the **False Attribution** class.
- **If** the predicted label is **False Attribution**:
 - **Output**: False Attribution
 - **End** the algorithm for this datapoint.

5. **Step 4:**

- **If** the pair does not belong to the **False Attribution** class, prompt the LLM to check if it belongs to the **Fabrication** class.
- **If** the predicted label is **Fabrication**:
 - **Output**: Fabrication
 - **End** the algorithm for this datapoint.

6. **Step 5:**

- **If** the pair does not belong to any of the above classes, classify it as **Correct**.

7. **Repeat** this procedure for every datapoint in the dataset.

6. Results

Table 1

Results of factual inaccuracy detection in Gujarati

Run	F1-Score	Rank
Run 1	0.0365	15
Run 2	0.0364	16
Run 3	0.0357	18
Run 4	0.0344	19
Run 5	0.0677	13

Table 2

Results of factual inaccuracy detection in Hindi

Run	F1-Score	Rank
Run 1	0.0653	17
Run 2	0.0364	18
Run 3	0.0357	19
Run 4	0.0344	21
Run 5	0.0677	16

Table 1 shows the results of factual inaccuracy in Gujarati. The results from the factual inaccuracy detection task in Gujarati reveal varying performance across five experimental runs, measured by F1 scores and their respective ranks. The F1 score is a key indicator of model accuracy, taking into account both precision and recall, and higher scores denote better performance.

Among the runs, Run 5 stands out with the highest F1 score of 0.0677, earning it a rank of 13th. This indicates that it was the most effective in identifying factual inaccuracies compared to the others. In contrast, Run 1 achieved a score of 0.0365 and ranked 15th, showing slightly better performance than Runs 2 and 3 but still significantly trailing behind Run 5.

Run 2 followed closely with a F1 score of 0.0364, ranking 16th, while Run 3 recorded a score of 0.0357 and ranked 18th, indicating a further decline in performance. Lastly, Run 4 had the lowest score at 0.0344, resulting in a rank of 19th, marking it as the least effective among all runs.

Overall, the results highlight that while there are minor differences in performance, none of the runs, except for Run 5, achieved satisfactory scores, which indicates ongoing challenges in developing effective models for detecting factual inaccuracies in Gujarati text.

Table 2 shows the results of factual inaccuracy in Hindi. The latest results from the factual inaccuracy detection task in Hindi reflect varying performance across five experimental runs, measured by their F1 scores and ranks.

Run 5 continues to lead with the highest F1 score of 0.0677, securing a rank of 16th, indicating it remains the most effective at detecting factual inaccuracies. Run 1 follows with a score of 0.0653, ranked 17th, showing a relatively strong performance and an improvement compared to its previous iteration.

In contrast, Run 2 recorded a F1 score of 0.0364 and is ranked 18th, representing a modest performance that is slightly better than Runs 3 and 4. Run 3 achieved a score of 0.0357, ranking 19th, indicating a minor decline in effectiveness compared to Run 2. Lastly, Run 4 had the lowest score of 0.0344 and is ranked 21st, marking it as the least effective among the runs.

Overall, these results suggest that while Run 5 maintains its position as the top performer, Run 1 has shown some improvement. However, the other runs struggle with lower scores, highlighting the ongoing challenges in effectively detecting factual inaccuracies in Hindi text.

7. Conclusion

In conclusion, the comparative analysis of factual inaccuracy detection in both Gujarati and Hindi demonstrates distinct performance trends among the experimental runs. In Gujarati, Run 5 emerges as the most effective model, achieving a F1 score of 0.0677, while the other runs exhibit significantly lower performance levels, with Run 4 lagging the furthest behind. Similarly, in Hindi, Run 5 retains its lead with a score of 0.0677, but Run 1 also shows notable improvement. For both Hindi and Gujarati, the ensemble approach shows the highest results. Despite these advancements, the overall scores across the runs indicate persistent challenges in developing robust models for detecting factual inaccuracies in both languages. The results underscore the need for further research and refinement to enhance the effectiveness of such detection systems in the future.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Drafting content, Grammar and spelling check, etc. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] Y. Li, Q. Li, L. Cui, W. Bi, Z. Wang, L. Wang, L. Yang, S. Shi, Y. Zhang, Mage: Machine-generated text detection in the wild, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 36–53.
- [2] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, O. M. Afzal, T. Mahmoud, G. Puccetti, T. Arnold, et al., Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection, arXiv preprint arXiv:2404.14183 (2024).
- [3] U. Hahn, I. Mani, The challenges of automatic summarization, Computer 33 (2000) 29–36.

- [4] T. B. Brown, Language models are few-shot learners, arXiv preprint arXiv:2005.14165 (2020).
- [5] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan, M. M. Kabir, A survey of automatic text summarization: Progress, process and challenges, IEEE Access 9 (2021) 156043–156070.
- [6] A. Das, H. Liu, V. Kovatchev, M. Lease, The state of human-centered nlp technology for fact-checking, Information processing & management 60 (2023) 103219.
- [7] K. Muthiah, Automatic Coherent and Concise Text Summarization using Natural Language Processing, Ph.D. thesis, Dublin, National College of Ireland, 2020.
- [8] C. van der Lee, A. Gatt, E. van Miltenburg, E. Krahmer, Human evaluation of automatically generated text: Current trends and best practice guidelines, Computer Speech & Language 67 (2021) 101151.
- [9] E. H. Hovy, C.-Y. Lin, L. Zhou, J. Fukumoto, Automated summarization evaluation with basic elements., in: LREC, volume 6, 2006, pp. 604–611.
- [10] K. S. Jones, Automatic summarising: factors and directions, arXiv preprint cmp-lg/9805011 (1998).
- [11] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004, pp. 74–81.
- [12] W. Kryściński, B. McCann, C. Xiong, R. Socher, Evaluating the factual consistency of abstractive text summarization, arXiv preprint arXiv:1910.12840 (2019).
- [13] A. Wang, K. Cho, M. Lewis, Asking and answering questions to evaluate the factual consistency of summaries, 2020. URL: <https://arxiv.org/abs/2004.04228>. arXiv:2004.04228.
- [14] Y. Huang, X. Feng, X. Feng, B. Qin, The factual inconsistency problem in abstractive text summarization: A survey, 2023. URL: <https://arxiv.org/abs/2104.14839>. arXiv:2104.14839.
- [15] R. Zhang, J. Ouni, S. Eger, Cross-lingual cross-temporal summarization: Dataset, models, evaluation, Computational Linguistics (2024) 1–44.
- [16] J. Wu, S. Yang, R. Zhan, Y. Yuan, D. F. Wong, L. S. Chao, A survey on llm-generated text detection: Necessity, methods, and future directions, 2024. URL: <https://arxiv.org/abs/2310.14724>. arXiv:2310.14724.
- [17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: <https://arxiv.org/abs/1810.04805>. arXiv:1810.04805.
- [18] 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>.
- [19] 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.
- [20] S. Satapara, P. Mehta, S. Modha, D. Ganguly, Key takeaways from the second shared task on indian language summarization (ILSUM 2023), in: K. Ghosh, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2023 - Forum for Information Retrieval Evaluation (FIRE-WN 2023), Goa, India, December 15-18, 2023, volume 3681 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 724–733. URL: <https://ceur-ws.org/Vol-3681/T8-1.pdf>.
- [21] 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.
- [22] 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.

- [23] 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, Gandhinagar, India. December 12-15, 2024, ACM, 2024.*
- [24] 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.
- [25] L. Wang, X. Chen, X. Deng, H. Wen, M. You, W. Liu, Q. Li, J. Li, Prompt engineering in consistency and reliability with the evidence-based guideline for llms, *npj Digital Medicine* 7 (2024) 41.
- [26] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, *ACM Computing Surveys* 55 (2023) 1–35.
- [27] X. Amatriain, Prompt design and engineering: Introduction and advanced methods, *arXiv preprint arXiv:2401.14423 (2024)*.
- [28] P. Srivastava, M. Malik, V. Gupta, T. Ganu, D. Roth, Evaluating llms’ mathematical reasoning in financial document question answering, in: *Findings of the Association for Computational Linguistics ACL 2024, 2024*, pp. 3853–3878.
- [29] L. Henrickson, A. Meroño-Peñuela, Prompting meaning: a hermeneutic approach to optimising prompt engineering with chatgpt, *AI & SOCIETY* (2023) 1–16.
- [30] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, S. Zhong, B. Yin, X. Hu, Harnessing the power of llms in practice: A survey on chatgpt and beyond, *ACM Transactions on Knowledge Discovery from Data* 18 (2024) 1–32.
- [31] L. Huang, S. Ma, D. Zhang, F. Wei, H. Wang, Zero-shot cross-lingual transfer of prompt-based tuning with a unified multilingual prompt, *arXiv preprint arXiv:2202.11451 (2022)*.
- [32] G. Xun, S. M. Land, A conceptual framework for scaffolding iii-structured problem-solving processes using question prompts and peer interactions, *Educational technology research and development* 52 (2004) 5–22.