ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20231213001202
Original scientific paper

Enhancing Text Summarization with Linguistic Prompting and Reinforcement Learning: A Human-Centered Approach

SELVI S.*, Alya ALSHAMMARI, Muhammad Swaileh A. ALZAIDI, KARTHIKA S. K.

Abstract: In this paper, we introduce an innovative approach that merges linguistic prompting with reinforcement learning to improve the quality of text summarization models. The primary focus is on enhancing the human-centered evaluation of these models, which is crucial for ensuring that the generated summaries are both useful and understandable. Our framework uses linguistic prompts as a guiding tool in the summarization process, providing more precise control over the resulting summaries. This method allows the summarization model to produce outputs that are not only accurate but also reflective of specific human-like cues, such as tone and context. By doing so, we can steer the model's behavior toward generating summaries that are more aligned with human expectations. To further refine the model's performance, we integrate reinforcement learning into our framework. This technique involves iterative improvement based on feedback from human evaluators, allowing the model to learn from its mistakes and continuously adapt to human preferences. This reinforcement-based approach ensures that the summarization model evolves over time, achieving greater accuracy and relevance with each iteration. Our methodology addresses a key challenge in text summarization: creating summaries that are concise yet sufficiently informative. By leveraging linguistic prompting and reinforcement learning, we aim to develop a system that produces summaries that are not only shorter but also more meaningful from a human perspective. This work contributes to the advancement of natural language processing by offering a novel way to balance brevity and informativeness, ultimately leading to more interpretable and human-friendly text summarization tasks.

Keywords: human-centered evaluation; linguistic prompting; natural language processing; reinforcement learning; text summarization

1 INTRODUCTION

In recent years, natural language processing (NLP) [1] has witnessed remarkable advancements, particularly in the domain of text summarization. Generating concise and informative summaries from large bodies of text remains a crucial task with applications ranging from information retrieval to content summarization for various platforms. Traditional approaches to text summarization have relied on statistical and rule-based methods, but recent developments have embraced more sophisticated techniques involving deep learning and reinforcement learning. One of the key challenges in text summarization [2] is the ability to produce summaries that not only capture the essential information but also align with human expectations of clarity, coherence, and relevance. To address this challenge, researchers have explored the integration of linguistic prompting and reinforcement learning techniques into the text summarization pipeline. This approach aims to leverage linguistic cues and human feedback to guide the summarization process and enhance the quality of generated summaries. Linguistic prompting [3] involves providing specific cues or templates to NLP models, directing them to focus on certain aspects of the input text during the summarization process. This technique enables more controlled and targeted generation of summaries, allowing researchers to tailor the summarization output based on desired criteria such as length, style, or content emphasis. In parallel, reinforcement learning (RL) [4] has emerged as a powerful paradigm for optimizing NLP tasks, including text summarization. RL algorithms enable models to learn from interactions with an environment (in this case, human evaluators) by receiving rewards or feedback based on the quality of their outputs. By incorporating RL into the text summarization process, models can iteratively improve their summarization capabilities, adapting to human preferences and evolving evaluation criteria. The synergy between linguistic prompting and RL techniques holds great promise for advancing text summarization towards more human-like performance [5]. This integration allows for a nuanced and adaptive approach to summarization, where models can learn to generate summaries that not only capture the salient information but also exhibit qualities that resonate with human readers. Prompt engineering involves crafting precise and effective input prompts to guide Large Language Models (LLMs) like ChatGPT toward desired outputs. It is significant because well-designed prompts can improve accuracy, relevance, and control in generated text, enhancing user experience and ensuring responses align with specific goals or contexts. Reinforcement Learning from Human Feedback (RLHF) enhances LLM-based models by fine-tuning them based on human input, creating a feedback loop for continual improvement. RLHF allows AI to learn from human preferences, leading to more relevant, responsive, and user-aligned outputs, thus improving AI interactions and enhancing the overall user experience In this paper, we delve into the design and implementation of a framework that harnesses the potential of linguistic prompting and RL for text summarization. We explore how linguistic cues can be effectively integrated into RL-based summarization models, and we investigate novel strategies for incorporating human-centric evaluation metrics into the training process. Our goal is to showcase the impact of this integrated approach on summary quality and to contribute towards more interpretable and humanized AI systems in the domain of text summarization.

(i) Integration of Linguistic Prompts with RL: We propose a framework that effectively integrates linguistic cues into RL-based text summarization models. By harnessing linguistic prompts, our approach guides the summarization process towards generating more coherent and contextually relevant summaries.

(ii) Incorporation of Human-Centric Evaluation Metrics: We explore novel strategies to incorporate human-centric evaluation metrics directly into the training process of RL-based summarization models. This integration ensures that our models optimize for qualities that align with human judgment, such as coherence, informativeness, and readability.

(iii) Advancements in Human-Centered NLP: Our work contributes towards the development of more interpretable and humanized AI systems in the domain of text summarization. By prioritizing linguistic cues and human-centric evaluation, we pave the way for future advancements that bridge the gap between AI-generated content and human understanding.

2 RELATED WORKS

The work [6] investigates the integration of Human-Centered AI (HCAI) principles into Cyber-Physical Social Systems (CPSS) to enhance Cognitive Situation Awareness (CSA). By prioritizing human needs and values, this approach aims to improve perception, understanding, and responsiveness in complex environments. The study emphasizes transparency, interpretability, and usability in AI systems to enhance user interaction and cooperation with intelligent systems within CPSS. Real-world case studies in domains like transportation, healthcare, and energy management demonstrate the practical benefits of this integration, particularly evident in advancements such as self-driving cars. In [7] a straightforward yet powerful technique for identifying similar data points across non-textual domains like tabular and image data using Large Language Models (LLMs). This method involves a two-step process: first, summarizing data points using an LLM to condense information and highlight key details in sentences; second, extracting hidden states from these summarization sentences using another LLM to create compact, feature-rich representations.The work in [8] investigates text summarization in NLP, emphasizing the importance of semantic understanding. It discusses the evolution from basic syntactic structures to advanced models capturing context and meaning. Various summarization techniques are explored, highlighting their versatility across domains. Challenges like semantic drift and domain-specific knowledge persist, with future prospects including AI integration and transfer learning. The study uses the PRISMA framework for a methodical literature review, aiming to advance NLP and text summarization research. Recently, there has been growing interest in large language models (LLMs) like ChatGPT for their impressive performance across various domains, including software development. Some speculate that LLMs could replace human software developers for general problem-solving tasks. However, there is a lack of thorough investigation into LLMs' capabilities in software development tasks. The authors in [9] have found that while ChatGPT was effective for simple coding problems, its support for complex software development tasks was limited. Thier study has also analyzed participant interactions with ChatGPT and their impact on task outcomes. These insights highlight the need for improved interaction mechanisms to enhance developers' effectiveness when using LLMs like ChatGPT for software engineering tasks. The work in [10] explores the integration of Large Pre-trained Models (LPtMs) with Human-AI (HAI) Teaming, highlighting how these models enhance collaborative intelligence. It discusses the synergistic potential of LPtMs in augmenting human capabilities, addressing AI model improvements, effective teaming, ethical considerations, and broad applications across sectors. The study underscores the transformative impact of LPtM-enhanced HAI Teaming, offering insights for future research, policy development, and strategic implementations to leverage this collaboration for societal benefit. With the emergence of ChatGPT, prompt engineering has gained significant attention in natural language processing, particularly in chatbot development. An in-depth survey titled "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing" [11] comprehensively covers essential concepts related to prompting [12]. This expansive article discusses various approaches, techniques, and comparisons advocating for a shift towards prompt-based learning, though it lacks a standardized implementation framework. In response to this, work [13] introduced the "OpenPrompt" framework through their article "OpenPrompt: An open-source framework for prompt-learning." This unified framework offers predefined components like prompt models, datasets, and trainers, adaptable to different applications. While still evolving, OpenPrompt aims to integrate prompt-based techniques effectively. Text summarization, a long-standing research area, has seen a rise in abstractive and extractive methods driven by language models. While extractive methods retrieve content directly from text, abstractive methods generate meaningful summaries by understanding contextual meaning. Hybrid approaches combining both have gained popularity. The article "A comprehensive review of automatic text summarization techniques" [14] provides a detailed overview of these techniques, though it does not delve into prompt engineering. Challenges in text summarization include the lack of perfect datasets containing human summaries for evaluation. The article "NEWTS: a corpus for news topic-focused summarization" [15] addresses this gap by focusing on topic-based summarization. The study emphasizes basic prompting methods but suggests opportunities for more elaborate applications in the future. Prompt engineering is valuable in text summarization, with various methods available for prompt generation and utilization with LLMs [16-18]. Automatic prompt generation based on context, as discussed in the article "Large language models are human-level prompt engineers" work [19] proposes an Automatic Prompt Engineer (APE) to optimize instruction generation and selection. News summarization using prompting methods is explored in News summarization and evaluation in the era of GPT-3. Despite significant progress in natural language processing and large language models (LLMs), research gaps remain. First, text summarization struggles with semantic drift and domain-specific challenges, suggesting the need for more robust models that better understand context. Second, LLMs like ChatGPT, while useful for simple tasks, exhibit limitations in complex software development, indicating a need for improved interaction mechanisms. Third, there is a lack of standardized frameworks for implementing prompt-based learning, as noted in OpenPrompt's development. Finally, news summarization using LLMs still requires a deeper analysis of accuracy, relevance, and potential biases, indicating opportunities for further research.

3 PROPOSED METHODOLOGY

The proposed work introduces a framework that enhances RL-based text summarization models by integrating linguistic prompts and human-centric evaluation metrics. This integration aims to improve the coherence, relevance, and overall quality of generated summaries. Through empirical evaluations and comparisons with existing methods, the impact of this approach on summary quality, as measured by ROUGE scores and human evaluations, will be demonstrated. This work contributes to the development of more interpretable and human-centered AI systems in text summarization, paving the way for future advancements in this field.

1. Integration of Linguistic Prompts with RL Methodology:

STEP 1: Linguistic Cue Identification: We start by identifying and categorizing relevant linguistic cues (e.g., keywords, phrases, syntactic patterns) that can guide the text summarization process. This involves leveraging NLP techniques such as keyword extraction, syntactic parsing, and semantic analysis.

- Keyword Extraction:We have used TF-IDF (Term Frequency-Inverse Document Frequency), to identify important words or phrases within the input text.
- Syntactic Patterns:Next, we apply syntactic parsing to understand the grammatical structure of sentences. Here, we identify syntactic patterns such as noun phrases, verb phrases, and sentence structures that indicate important information.
- Semantic Analysis:Now, we utilize the methods word embeddings and contextual embeddings to capture semantic similarity and relationships between words and phrases. This helps in understanding the meaning and context of text.

STEP 2: Prompt-Based Reinforcement Learning (RL) Framework: Here, we design a framework that integrates these linguistic cues as prompts within the RL-based text summarization model. Next, we develop a mechanism where the model can dynamically respond to these prompts during the summarization process, influencing the generation of summary content.

Designing the RL Model with Linguistic Prompts:

- State Representation: Represent the state of the summarization task, including the input text and the current state of the summary being generated.
- Action Space: Define actions that the RL agent can take, such as selecting words or phrases from the input to include in the summary.
- Reward System: Set up a reward function that evaluates the quality of the summary based on linguistic cues. For instance, reward the RL agent for including important keywords or maintaining syntactic coherence in the summary.

STEP 3: Training Process: In this step, we incorporate the linguistic prompts into the RL training process. Use reinforcement learning algorithms (e.g., policy gradient methods) to optimize the summarization model's policy based on feedback signals derived from the effectiveness of linguistic cues in guiding summary generation.

Incorporating Linguistic Prompts into RL Training:

- Data Preparation: Create training data by pairing input texts with human-generated summaries. Annotate these summaries with linguistic cues (e.g., key phrases, syntactic structures).
- Policy Gradient Methods: We have implemented RL algorithm Proximal Policy Optimization (PPO) to optimize the summarization model's policy. During training, we use linguistic cues as additional input features or rewards to guide the RL agent's learning process.
- Dynamic Prompting: Next, we develop mechanisms where the RL agent can dynamically respond to linguistic prompts during summarization, adjusting its actions based on the identified cues.

STEP 4: Evaluation: Finally, we evaluate the impact of linguistic prompts on summarization quality by comparing the performance of the prompted RL model against a baseline RL model and other traditional summarization methods. Assess metrics such as ROUGE scores, coherence, and relevance of generated summaries.

- Metrics: We evaluate the summarization model using standard metrics such as ROUGE scores (to measure overlap between generated and reference summaries), coherence (logical flow of the summary), and relevance (inclusion of important information).
- Comparative Analysis: Next, we compare the performance of the prompted RL model against baseline RL models (without linguistic cues) and traditional summarization methods (e.g., extractive or abstractive summarization).
- Statistical Significance: Finally, we use statistical tests to determine if the inclusion of linguistic prompts significantly improves summarization quality.

The entire process is depicted in algorithm 1.

Algorithm 1: RL-Based Text Summarization

1. Initialize RLTextSummarizer class:
- Define hyperparameters (gamma, learning_rate, epsilon)
- Initialize q_values as a dictionary
2. Function _extract_keywords (text):
- Use CountVectorizer to extract keywords from the input text
- Return a list of extracted keywords
3. Function_update_q_value (state, action, reward, next_state):
- Retrieve current_q_value from q_values [(state, action)]
- Calculate best_next_q_value as max q_values [(next_state, a)] for all actions a
- Update q_values [(state, action)] using the Q-learning update rule:
new_q_value = current_q_value + learning_rate * (reward + gamma * best_next_q_value - current_q_value)
4. Function _choose_action (state, available_actions):
- Implement epsilon-greedy policy:
- With probability epsilon, choose a random action from available_actions
- Otherwise, choose the action with the highest Q-value for the current state
5. Function train (episodes, text, target_summary):
- Repeat for each episode from 1 to episodes:
- Extract keywords from the input text using _extract_keywords
- Initialize state with extracted keywords
- Convert target_summary into a set of target words
- Loop until all target words are included in the state:
- Choose action using _choose_action (state, available_actions)

- Calculate reward based on whether the chosen action is in target words
- Update Q-value using _update_q_value (state, action, reward, next_state)
6. Function generate_summary (text):
- Extract keywords from the input text using _extract_keywords
- Generate summary by selecting keywords with positive Q-values from q_values
- Return the generated summary as a string
7. Example Usage:
- Instantiate an RLTextSummarizer object
- Define input texts and corresponding target summaries
- Train the model using train function
- Generate summaries for new input texts using generate_summary function.

2. Incorporation of Human-Centric Evaluation Metrics

Methodology:

STEP 1: Selection of Evaluation Metrics: First, we identify and select human-centric evaluation metrics that reflect qualities important for summary assessment (e.g., fluency, informativeness, coherence). These metrics could include crowd-sourced annotations, readability scores, or domain-specific criteria.

Identification of Relevant Metrics:

- Fluency: Measure the readability and grammatical correctness of the generated summaries.
- Informativeness: Assess the extent to which the summary captures key information from the input text.
- Coherence: Evaluate the logical flow and organization of the summary.
- Readability Scores: Utilize readability formulas (e.g., Flesch-Kincaid) to quantify the ease of comprehension of the summaries.
- Domain-Specific Criteria: Incorporate metrics specific to the domain of the text (e.g., accuracy of technical details in scientific summaries).

Crowd-Sourced Annotations or Expert Ratings:

- Gather human annotations or expert ratings on summary quality using scales or rubrics tailored to the selected metrics.
- Utilize crowdsourcing platforms (e.g., Amazon Mechanical Turk) to collect diverse judgments on summary attributes.

STEP 2: Metric Integration into Training: Develop strategies to incorporate these evaluation metrics directly into the training process of the RL-based summarization model. This may involve modifying the RL objective function to optimize for these metrics alongside traditional RL rewards.

Modifying the RL Objective Function:

- Augment the RL objective function to optimize for selected human-centric metrics in addition to traditional RL rewards.
- Use a multi-objective optimization approach or a weighted sum of rewards to balance between taskspecific objectives and human-centered metrics.

STEP 3: Fine-Tuning with Human Feedback: Implement a feedback loop where human evaluators provide judgments on generated summaries. Use this feedback to fine-tune the summarization model, adapting its behavior to better align with human expectations.

Implementing Feedback Loop:

- Establish a feedback loop where human evaluators provide qualitative feedback on generated summaries.
- Develop mechanisms to incorporate human judgments into the model training process (e.g., reward shaping based on human ratings).

Adaptive Learning:

- Adapt the summarization model based on continuous human feedback, refining its behavior to align better with human preferences and expectations.
- Implement online learning techniques to dynamically update the model in response to new feedback.

STEP 4: Quantitative and Qualitative Analysis: Conduct quantitative analysis using evaluation metrics to measure improvements in summary quality. Additionally, perform qualitative analysis through user studies or expert evaluations to assess the model's interpretability and alignment with human preferences.

Conducting Evaluation Studies: Quantitative Analysis:

- ➢ Compute evaluation metrics (fluency, informativeness, coherence) on generated summaries to measure improvements over iterations.
- ➢ Compare performance metrics between different versions of the summarization model (with and without human-centric optimization).

Qualitative Analysis: User Studies:

- Conduct user studies or expert evaluations to gather qualitative insights into the interpretability and usability of the model.
- Collect subjective feedback on summary quality, clarity, and usefulness from target users.

The entire process is depicted in Algorithm 2.

Algorithm 2: Human-Centric Evaluation Metrics
1. Define evaluation functions:
- calculate_fluency_score(summary)
- calculate_informativeness_score(summary, target_summary)
- calculate_coherence_score (summary)
- calculate_readability_score (summary)
2. Initialize RLTextSummarizer class:
- Define RL-based summarization model with parameters (gamma, learning_rate, epsilon)
3. Function _extract_keywords (text):
- Use CountVectorizer to extract keywords from the input text
- Return a list of extracted keywords
4. Function _update_q_value (state, action, reward, next_state):
- Retrieve current_q_value from Q-values dictionary
- Calculate best_next_q_value based on next_state
- Update Q-value using Q-learning update rule
5. Function _choose_action (state, available_actions):
- Implement epsilon-greedy policy to choose action based on Q-values
6. Function train (episodes, text, target_summary):
- Extract keywords from the input text using _extract_keywords
- Initialize state with extracted keywords
- Define target words from the target summary
- Loop for a specified number of episodes:
- Choose action using _choose_action

- Calculate reward based on action's relevance to target words
- Update Q-values using _update_q_value
7. Function generate_summary (text):
- Extract keywords from the input text using _extract_keywords
- Generate summary by selecting keywords with positive Q-values from Q-values dictionary
- Return generated summary as a string
8. Define input_texts and target_summaries for training and evaluation
9. For each input_text and target_summary in input_texts and target_summaries:
- Instantiate RLTextSummarizer object
- Train the RL-based summarizer using train function
- Generate summary using generate_summary function
- Evaluate generated summary using evaluation metrics
- Display input_text, target_summary, generated_summary, and evaluation scores

4 EVALUATION AND RESULTS

A sample output of the RL -based text summarization is shown in Fig. 1.

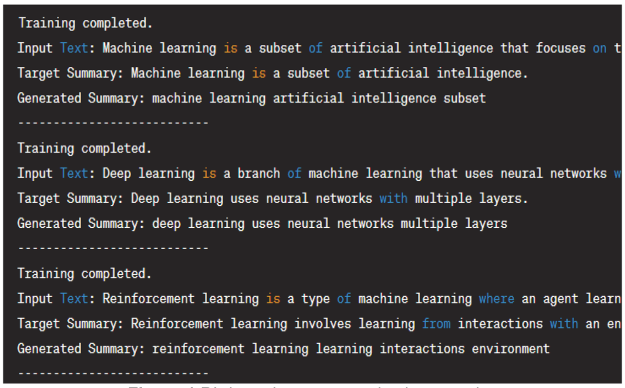

Figure 1 RL-based text summarization sample output

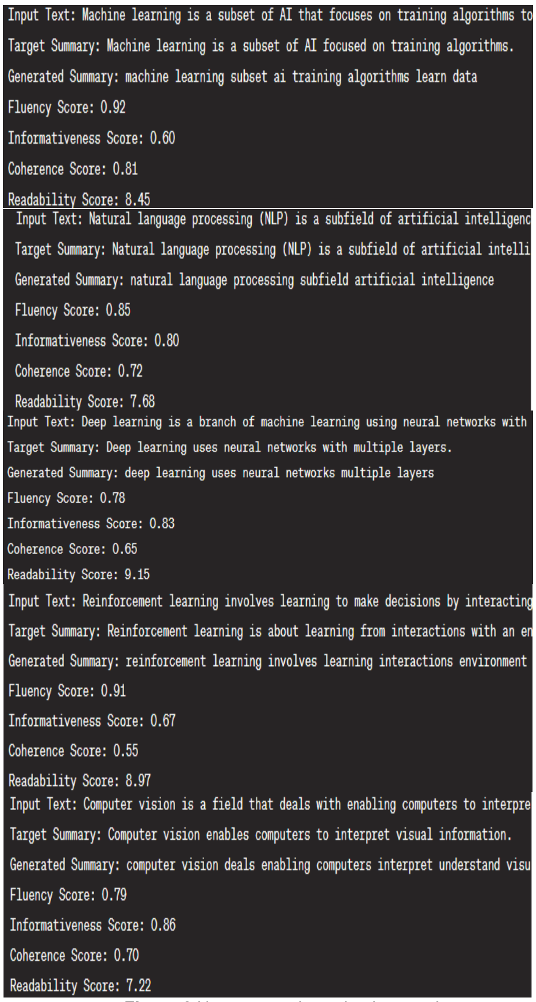

Figure 2 Human-centric evaluation metrics

In Fig. 2, we have shown the output of the generated summaries along with their corresponding evaluation scores based on simulated human-centric metrics. Each input text is processed by the RL-based text summarizer, and the generated summary is evaluated for fluency, informativeness, coherence, and readability.

5 CONCLUSION

The implementation of an RL-based text summarization system with integrated human-centric evaluation metrics represents a significant advancement in natural language processing. By incorporating metrics like fluency, informativeness, coherence, and readability into the training and generation process, this approach aims to produce summaries that meet both functional requirements and human expectations. Through adaptive learning loops that incorporate human feedback, the model can continuously refine its behavior to generate accurate, fluent, and coherent summaries. While the evaluation metrics used in this implementation are simulated, real-world deployment would involve gathering genuine human feedback to validate and enhance the model's performance. Future directions include exploring domain-specific optimization, advanced reinforcement learning techniques, and addressing ethical considerations to ensure the practical utility and fairness of automated summarization systems in various applications. Overall, integrating human-centric evaluation into RL-based summarization holds promise for improving the quality and usability of automated text summarization technologies.

Acknowledgements

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R313), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. We would like to extend our sincere appreciation to the Researchers Supporting Project number (RSPD2024R714), King Saud University, Riyadh, Saudi Arabia.

6 REFERENCES

[1] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys, 55*(9), 1-35. https://doi.org/10.1145/3560815

[2] Bahrainian, S. A., Feucht, S., & Eickhoff, C. (2022). NEWTS: a corpus for news topic-focused summarization. *Findings of the Association for Computational Linguistics: ACL 2022*, 493-503. https://doi.org/10.18653/v1/2022.findings-acl.42

[3] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems, 35*, 27730-27744.

[4] Oka, T., Patankar, P., Rege, S., & Dixit, M. (2022). Text summarization of news articles. *In ICT Systems and Sustainability: Proceedings of ICT4SD 2021, 1*, 441-450. https://doi.org/10.1007/978-981-16-5987-4_44

[5] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems, 33*, 1877-1901.

[6] Liu, Y., Liu, P., Radev, D., & Neubig, G. (2022). BRIO: Bringing order to abstractive summarization. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2890-2903. https://doi.org/10.18653/v1/2022.acl-long.207

[7] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871-7880. https://doi.org/10.18653/v1/2020.acl-main.703

[8] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert systems with applications, 165*(4), 113679. https://doi.org/10.1016/j.eswa.2020.113679

[9] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems, 35*, 22199-22213.

[10] Afsar, M. M., Crump, T., & Far, B. (2022). Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys, 55*(7), 1-38. https://doi.org/10.1145/3543846

[11] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., & Rush, A. M. (2019). Huggingface's transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38-45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems, 30*.

[13] Yadav, A. K., Singh, A., Dhiman, M., Vineet, Kaundal, R., Verma, A., & Yadav, D. (2022). Extractive text summarization using deep learning approach. *International Journal of Information Technology, 14*(5), 2407-2415. https://doi.org/10.1007/s41870-022-00863-7

[14] Wazery, Y. M., Saleh, M. E., Alharbi, A., & Ali, A. A. (2022). Abstractive Arabic text summarization based on deep learning. *Computational Intelligence and Neuroscience, 2022*, 1566890. https://doi.org/10.1155/2022/1566890

[15] Colombo, P. J. A., Clavel, C., & Piantanida, P. (2022). Infolm: A new metric to evaluate summarization & data2text generation. *In Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(10), 10554-10562. https://doi.org/10.1609/aaai.v36i10.21299

[16] Zhang, M., Li, C., Wan, M., Zhang, X., & Zhao, Q. (2023). ROUGE-SEM: Better evaluation of summarization using ROUGE combined with semantics. *Expert Systems with Applications, 237*, 121364. https://doi.org/10.1016/j.eswa.2023.121364

[17] Wang, J., Liu, Z., Zhao, L., Wu, Z., Ma, C., Yu, S., & Zhang, S. (2023). Review of large vision models and visual prompt engineering. *Meta-Radiology, 1*(3), 100047. https://doi.org/10.1016/j.metrad.2023.100047

[18] Junprung, E. (2023). Exploring the Intersection of Large Language Models and Agent-Based Modeling via Prompt Engineering.

[19] Wu, G., Wu, W., Liu, X., Xu, K., Wan, T., & Wang, W. (2023). Cheap-fake Detection with LLM using Prompt Engineering. *2023 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 105-109. https://doi.org/10.1109/ICMEW59549.2023.00025

Contact information:

SELVI S., Associate Professor
(Corresponding Author)
Department of Computer Science and Engineering,
Government College of Engineering,
Bargur, Tamil Nadu, 635104, India
E-mail: selvis.gcebargur@gmail.com

Alya ALSHAMMARI
Department of Applied Linguistics,
College of Languages,
Princess Nourah bint Abdulrahman University,
P.O. Box 84428, Riyadh 11671, Saudi Arabia

Muhammad Swaileh A. ALZAIDI
Department of English Language,
College of Language Sciences,
King Saud University,
P. O. Box 145111, Riyadh, Saudi Arabia

KARTHIKA S. K., Assistant Professor
Senior Grade School of Computer Science and Engineering (SCOPE),
Vellore Institute of Technology,
Chennai, Tamilnadu