CRF^2: Context Reasoning Faithful Framework

Hugo Firmino Damasceno[1], **Leonardo Sampaio Rocha**[1]

Graphs and Computational Intelligence Lab - LAGIC, Postgraduate Program in Computer Science - PPGCC, Center for Science and Technology - CCT, State University of Ceará - UECE, Fortaleza, Ceará, Brazil.

hugo.damasceno@aluno.uece.br, leonardo.sampaio@uece.br

Abstract. *Large Language Models (LLMs) excel in tasks like text generation and question answering, but often rely on parametric knowledge, leading to hallucinations when required to reason strictly within a given context. In this paper, we propose a conceptual framework for context-faithful reasoning, ensuring responses are grounded solely in the provided information. We evaluate our approach using the RealTime QA dataset, which features open-domain, time-sensitive questions that require grounding in a specific document. Our experiments demonstrate that the proposed method outperforms existing prompt engineering techniques on abstention tasks, achieving an accuracy of 0.95 compared to baseline models.*

1. Introduction

The growing interest in Large Language Models (LLMs) has driven research across various domains, given their ability to handle complex tasks. The widespread adoption of models such as ChatGPT [OpenAI 2022] has further expanded their applications. However, when used without proper guidance in instruction formulation, these systems can generate inaccurate responses, with distorted concepts or outdated information [Kasai et al. 2024], particularly in tasks that require reasoning.

To mitigate these limitations, prompt engineering, a set of techniques focused on the strategic design of instructions, has emerged as an effective approach, enabling the guidance of model outputs without the need for parameter tuning [Saravia 2022] [Sahoo et al. 2024]. Despite significant progress, challenges remain, especially in tasks involving strategic reasoning and multi-step inference [Yao et al. 2023]. Techniques such as Chain of Thought (CoT) [Wei et al. 2023], designed to address this type of task, do not always yield consistent performance gains [Bao et al. 2024], and in some cases, their impact is limited.

Recent studies indicate that LLMs tend to operate in a slow and deliberate manner, which may hinder their effectiveness in cumulative reasoning tasks [Zhang et al. 2024]. Furthermore, [Pan et al. 2024] identify two key challenges: faithless generation, where responses fail to reflect specific or up-to-date facts, and weak reasoning, in which models struggle to properly analyze the given context, undermining the usefulness of their responses. While parametric knowledge, that is, the information stored in the model's parameters during training [Xu et al. 2024], is valuable for general knowledge tasks, its dominance can impair performance in scenarios that demand strict adherence to the provided context [Zhou et al. 2023].

A promising direction for improving performance in reasoning tasks is the incorporation of auxiliary context. However, studies show that LLMs often partially or entirely disregard this context, posing a significant challenge for applications that require consistency and contextual fidelity. Ensuring such fidelity requires that the model infer exclusively from the provided information, avoiding trivial assumptions or biases derived from prior knowledge.

In this work, we propose a framework that guides LLMs to reason exclusively based on the provided context, eliminating reliance on external knowledge or additional training adjustments.

2. Overview

Large Language Model (LLM) Language models are probabilistic models designed to represent and predict the structure of a natural language. Essentially, these models estimate the probability of a sequence of words occurring based on a given context [Xiao and Zhu 2025]. Let $x_0, x_1, \ldots, x_m$ be a sequence of words from a vocabulary ν, where x_0 is the start symbol $\langle s \rangle$. The probability of this sequence can be defined using the chain rule:

$$\begin{aligned} Pr(x_0, ..., x_m) &= Pr(x_0) \cdot Pr(x_1|x_0) \cdot Pr(x_2|x_0, x_1) \cdots Pr(x_m|x_0, ..., x_{m-1}) \\ &= \prod_{i=0}^{m} Pr(x_i|x_0, ..., x_{i-1}) \end{aligned} \tag{1}$$

where $Pr(x_i \mid x_0, \ldots, x_{i-1})$ represents the probability of token x_i occurring given the history of previous tokens $(x_0, \ldots, x_{i-1})$. These probabilities reflect how likely it is that a given sequence will be generated, considering the order and context of the tokens. Tokens are the smallest units of text processed by the model, they may correspond to an entire word, part of a word, an individual character, or even a symbol.

Architecture The architecture of a language model is based on a neural network known as the Transformer, whose structure relies on the self-attention mechanism. This mechanism enables the model to evaluate relationships between all tokens in a sequence, regardless of their distance, thereby facilitating the capture of contextual dependencies [Vaswani et al. 2023]. As a result, the model learns the relationships among tokens across long text sequences, allowing it to understand the meaning of each token within its contextual setting [Jurafsky and Martin 2025].

Training According to [Devlin et al. 2019], training large language models typically involves two main stages: pre-training and fine-tuning. Pre-training is the initial phase, carried out in an unsupervised manner, in which the model is exposed to a large textual corpus, usually composed of diverse sources such as books, articles, and web pages. The goal of this stage is to enable the model to learn general language patterns, including grammar, semantic structure, and world knowledge [Jurafsky and Martin 2025]. The most common approaches used during pre-training are causal language modeling and masked language modeling. In causal language modeling, the model learns to predict the next word in a sequence. In masked

language modeling, the model receives input sentences with hidden (masked) words and must predict the missing terms. After pre-training, the model can be adapted to specific tasks through fine-tuning, which occurs in a supervised setting. In this stage, the model is trained on labeled datasets to optimize its performance on a particular application. This work does not aim to train a model from scratch, nor to perform retraining or fine-tuning. Instead, it seeks to evaluate the capability of pre-trained models to perform tasks that require comprehension and reasoning based on a given context.

Prompt Engineering A prompt is an instruction given to a language model to guide it in performing a specific task [Huyen 2025]. Well-crafted prompts can significantly improve the accuracy and relevance of responses while reducing ambiguity and minimizing the risk of hallucinations [Saravia 2022]. The practice of designing these prompts strategically is known as prompt engineering, which has emerged to enhancing the performance of large language models (LLMs) [Jurafsky and Martin 2025].

3. Related Work

To identify the most relevant studies related to reasoning faithfulness in large language models (LLMs), we conducted a systematic search using the query: (faithful) AND (llm OR (large AND language AND model)) AND (context) AND (reasoning). The search covered the period from 2021 to 2025. Below, we discuss the studies that are most closely aligned with the proposed framework.

In [Zhou et al. 2023], addresses the issue of faithfulness in language models for context-specific Natural Language Processing (NLP) tasks. In this context, faithfulness refers to the model's ability to infer information solely from the provided context, without relying on memorized knowledge or statistical biases. The authors identify two main challenges: knowledge conflict and abstention prediction. To address these issues, the researchers propose prompting strategies aimed at improving model faithfulness. Among these strategies are opinion-based prompts, which reframe the context as a narrator's statement and encourage the model to focus on the given text; instructional prompts, which explicitly guide the model to ground its answers in the context through clear instructions; and counterfactual demonstrations, which present examples with altered information to prompt the model to update its predictions based on the context, thereby reducing reliance on memorized knowledge.

The study by [Pan et al. 2024], introduces a new framework called Chain-of-Action (CoA), designed to enhance the ability of large language models (LLMs) to answer multimodal questions in a reliable and well-grounded manner. The main goal of CoA is to address two common challenges in current Question Answering (QA) approaches: (i) unfaithful hallucinations, where the model generates responses that are inconsistent with real-world or domain-specific facts, and (ii) poor performance in reasoning over compositional information. The core contribution of the work is a novel retrieval and reasoning mechanism that decomposes complex questions into a structured sequence of actions. This approach is implemented through systematic prompts and predefined actions that guide the LLM through the process of searching, verifying, and inferring information. To support this, the authors propose

three types of plug-and-play actions that can be adapted to various domains and enable real-time retrieval from heterogeneous sources. Additionally, they introduce the Multireference Faithfulness Score (MRFS), a verification system that detects and resolves conflicts between LLM-generated answers and retrieved information, thereby enhancing the reliability of the responses.

[Li et al. 2024] analyzes how memory strength and the style of evidence presentation influence the contextual faithfulness of large language models (LLMs) when using external information during answer generation. The authors propose a metric to quantify an LLM's memory strength based on the consistency of its responses to different paraphrases of the same question. Using the PopQA and Natural Questions datasets, along with models such as LLaMA2, ChatGPT, and GPT-4, the experiments reveal that the stronger a model's memory, the more it tends to rely on its internal knowledge. However, presenting evidence in a paraphrased form significantly increases the LLMs' receptiveness to external information, thereby improving their faithfulness to the given context.

The study [Huang et al. 2024], proposes the Prompting Explicit and Implicit knowledge (PEI) framework, which aims to align the reasoning of pre-trained language models (PLMs) with the human cognitive process in multi-hop question answering (QA) tasks. Inspired by studies in reading psychology, PEI models explicit knowledge as the passages provided in the input and implicit knowledge as the information acquired during the model's pre-training, using unified prompts to integrate both types of knowledge. The approach also incorporates question-type-specific reasoning and enables filtering of irrelevant information from the passages. Experiments on the HotpotQA dataset show that PEI achieves performance comparable to state-of-the-art methods, with ablation studies confirming the effectiveness of integrating explicit and implicit knowledge.

To better highlight the distinct contribution of the proposed framework, table 1 presents a comparative summary of the main approaches discussed. While methods such as CoA and PEI focus on multimodal or multi-hop reasoning, this work uniquely integrates structured verification and contextual validation in a sequential pipeline, prioritizing abstention when information is insufficient. Moreover, unlike works that rely heavily on memory strength mitigation, our framework emphasizes context-based reasoning and response verification strictly within the provided information.

Table 1. Comparison of related works

Study	**Objective**	**Prompt Strategy**
[Zhou et al. 2023]	Improve context faithfulness via prompting	Opinion + Instruction
[Pan et al. 2024]	Faithful multimodal QA with structured decomposition	Chain-of-Action prompts
[Li et al. 2024]	Analyze influence of memory strength on context faithfulness	Paraphrased evidence injection
[Huang et al. 2024]	Integrate explicit + implicit knowledge in multi-hop QA	Unified prompts with question types
This work	Context-based reasoning	Multi-phase prompting

4. Context Reasoning Faithful Framework

Inspired by reading and text interpretation strategies identified in cognitive science, we propose CRF^2, a framework designed to guide language models in generating responses that remain faithful to the provided context. CRF^2 consists of four main stages, as illustrated in Fig. 1. The first three stages are based on techniques well-established in the literature, while the final stage introduces a strategy for validating the contextual fidelity of the response. Each stage is described in detail below.

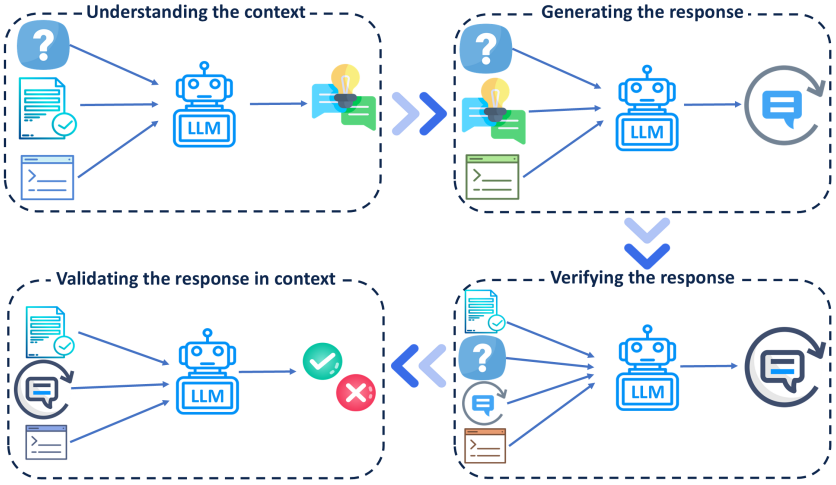

Figure 1. The diagram presents the four steps of the framework: (i) *Understanding the context*, where the LLM analyzes the question, the document; (ii) *Generating the response*, where the LLM uses the understood context to generate possible answers; (iii) *Verifying the response*, a step in which the generated answer is re-evaluated; and (iv) *Validating the response in context*, where the LLM confirms, based on the original content, whether the final answer is in accordance with the context provided.

Understanding the context (UC) is based on the approach proposed by [Yu et al. 2024], in which the model is guided to generate structured summaries of the context, highlighting the most relevant information for answering the question. This strategy aims to filter pertinent information and disregard irrelevant or noisy segments, thereby reducing the likelihood of the model producing inaccurate or hallucinated responses. To achieve this, the model receives three inputs: (i) the question, which defines the problem to be solved and may require inference from the context; (ii) the context, which gathers the information that supports the answer; and (iii) the prompt, which provides instructions to guide the model in understanding the question and extracting the necessary evidence.

Prompt UC:
Read the text below carefully and then analyze the question provided. Your goal in this step is to:

- Understand the general content of the text.
- Identify exactly what the question is asking.
- Find the most relevant passage in the text to answer the question.
- If you find a relevant passage, provide the exact quote.
- If no part of the text provides a direct or indirect answer, simply respond with, 'I don't know.'

Text: {context}
Q: {question}
A: {options}

Generating the response (GR) is the stage in which the language model constructs a grounded response based on the summaries extracted in the previous phase. Unlike conventional approaches that rely on direct answer generation, this stage follows a step-by-step reasoning process[Wei et al. 2023], ensuring that the response is built logically upon the extracted summaries. The model receives the following inputs: (i) the question, which defines the problem to be solved; (ii) the context summaries, which are the structured segments previously extracted from the original context. These summaries contain filtered information that may be useful for answering the question; and (iii) the prompt, which provides instructions for the model to reason sequentially, validating each step before reaching a final conclusion. The prompt also instructs the model to explicitly acknowledge when the provided summaries lack sufficient information to answer the question, thereby avoiding speculation or unsupported responses.

Prompt GR:
You will be given notes and a question. The notes contain information to help you answer the question. If the notes do not provide the answer, choose 'I don't know'. Otherwise, reason step by step and select the correct option.
Quot: {quot}
Q: {question}
A: {options}

Verifying the response (VR) is the stage in which the language model adopts a strategy inspired by [Dhuliawala et al. 2023], where it autonomously generates a set of questions aimed at assessing the correctness of the initial answer by identifying potential inconsistencies, misinterpretations, or reasoning flaws. This verification process is structured into three steps: (1) generation of verification questions, (2) analysis of the initial answer based on the responses to the verification questions, and (3) generation of a revised version of the answer if any inconsistencies are detected. The inputs for this stage are: (i) Context: the information that must be used in the answer; (ii) Question: the original problem to be solved; (iii) Generated Answer: the output produced in the answer generation phase; and (iv) Prompt: explicit instructions that guide the verification process.

Prompt VR:
You will be given a text, a question and an answer. Your goal is to check the answer:

- Generate a list of check questions that can help you self-analyze whether there are any errors in the original answer.
- Answer each check question in turn and then check the answer against the original answer for inconsistencies or errors.
- Given the inconsistencies discovered (if any), generate a revised answer incorporating the results of the check.
- If there are no inconsistencies, simply reaffirm the original answer as correct.

Text: {context}
Q: {question}
A: {options}
Answer: {answer GR}

Validating the response in context (VRC) aims to ensure that the generated answer is fully grounded in the provided context, without introducing external or unsupported information. Unlike the previous verification stage, which focuses on the internal consistency of the answer, this phase evaluates how well the response aligns with the original context, assessing to what extent the presented information is directly derived from the input data. The adopted strategy involves two main steps: the model identifies which parts of the answer can be directly mapped to the context and which parts represent additional inferences. If the answer includes information that cannot be justified by the context, the model must detect these elements and either revise the answer to remain strictly within the provided content or state that it does not have enough information to respond. The model receives the following inputs for this stage: (i) Context: the original set of information used as the basis for the answer; (ii) Final Answer: the revised output from the previous stage; and (iii) Prompt: instructions guiding the context-based verification process.

Prompt VRC:
You will receive a text, a question and an answer. Your goal is to:

- Find a passage in the text that logically justifies the answer.
- If you find one, cite the passage and explain the logical relationship between the passage and the answer.
- If you do not find a passage that justifies the answer, change the answer to ”I don’t know”.

Text: {context}
Q: {question}
A: {options}
Answer: {answer VR}

While both the VR and VRC stages aim to enhance the reliability of the generated response, they operate with distinct verification objectives. The VR stage focuses on the internal consistency of the answer by analyzing whether the reasoning

is coherent and free from contradictions or misinterpretations, regardless of whether the information aligns with the given context. In contrast, the VRC stage is concerned with contextual fidelity, ensuring that all parts of the answer are explicitly supported by the provided context, avoiding hallucinations or the inclusion of external knowledge. Thus, while VR evaluates the plausibility of the reasoning, VRC verifies the groundedness of the content.

5. Experiments and Results

We conducted two experiments using datasets specifically designed to evaluate contextual faithfulness. The first experiment requires the model to find the answer strictly within the provided context, and if no sufficient information is found, it must respond with "I don't know." The second experiment uses data with counterfactual contexts, which include statements that contradict common sense or widely accepted knowledge. This setting directly challenges the model's parametric knowledge by confronting it with conflicting contextual information.

Benchmarks For the first experiment, we used the RealTime QA dataset[Kasai et al. 2024], a benchmark created to test language models' ability to answer questions based on recent and continuously evolving events—that is, questions whose answers cannot be known solely from the model's prior knowledge. To enable result comparison with previous work, we adopted a modified version of RealTime QA, introduced by [Zhou et al. 2023], consisting of 113 instances. Of these, 63 have explicit answers in the retrieved documents, while 50 are questions whose answers cannot be inferred from the given context. To handle these cases, the authors added an additional answer choice: "E) I don't know." Based on this, two evaluation classes were defined: Class 1: when the model selects "I don't know" upon finding no supporting evidence in the context and Class 0: when the model selects any other answer despite the absence of contextual support.

The second experiment was conducted using the FaithEval benchmark [Ming et al. 2025], designed to assess the factual consistency of answers generated by language models. This dataset includes three distinct types of context: Unanswerable Context, the context does not contain the answer to the question, Inconsistent Context, multiple answers are supported by different documents and Counterfactual Context – the context contains counterfactual statements that contradict common sense or world knowledge. In this study, we initially focused only on the counterfactual context setting to evaluate how the model handles conflicts between the context and its internal knowledge.

Model The experiments used two models: LLaMA 3 70B [Minaee et al. 2024], chosen for its competitiveness with state-of-the-art models and open-source availability, and **GPT-3.5 Turbo** (version 0125) [OpenAI 2022], widely adopted in the literature. Both were configured with temperature set to 0, a maximum of 2048 tokens, and default values for all other parameters, ensuring deterministic outputs without context truncation.

Evaluation methods The performance evaluation was conducted based on metrics designed for contextual answering scenarios with the possibility of abstention, as proposed by [Zhou et al. 2023]. Three main metrics were used: HasAns, which

measures accuracy on instances where the retrieved documents contain the answer; NoAns, which assesses the model's ability to correctly abstain by selecting the "I don't know" option when the context lacks sufficient information; and All, which corresponds to the overall accuracy across all test instances.The performance evaluation was carried out using standard machine learning metrics: F1-score, recall, accuracy, and precision. Accuracy was included due to its widespread use in studies assessing contextual faithfulness. The F1-score, as the harmonic mean of precision and recall, provides a more comprehensive view of the framework's overall performance. Below is the prompt used in [Zhou et al. 2023] for comparison.

> Instruction: answer a question based on the provided input-output pairs.
> Bob said: {context}
> In Bob's opinion, {question}
> Choice: {options}

Table 2 presents the accuracy results obtained from the experiments using the Realtime QA benchmark, comparing different configurations of the proposed framework (CRF²) with the Opin+Instruc baseline. The results are shown for both GPT-3.5 Turbo and LLaMA 3 70B models. For GPT-3.5, the baseline Opin+Instruc achieved 0.95 accuracy for answerable questions (HasAns), but dropped significantly to 0.56 for unanswerable ones (NoAns), resulting in an overall accuracy of 0.77. In contrast, the full CRF² pipeline (UC+GR+VR+VRC) improved NoAns detection (0.90). With LLaMA 3 70B, all CRF² configurations outperformed the baseline across all metrics, with the full pipeline achieving 0.95 overall accuracy, demonstrating high fidelity to the context and balanced handling of both answerable and unanswerable questions.

Table 2. Comparison accuracy results between CRF² and the context-faithful approach. Realtime QA Benchmark

Method	GPT 3.5			Llama3 70B		
	HasAns	NoAns	All	HasAns	NoAns	All
Opin+Instr [Zhou et al. 2023]	0.95	0.56	0.77	0.94	0.90	0.92
UC	0.76	0.82	0.79	0.95	0.94	0.95
UC+GR	0.97	0.82	0.90	0.95	0.94	0.95
UC+GR+VR	0.95	0.76	0.87	0.95	0.94	0.95
UC+GR+VR+VRC	0.65	0.90	0.76	0.95	0.94	0.95

The Fig 2 shows the results of the experiment using the FaithEval benchmark, with outcomes again presented by framework stage. In this experiment, we maintained the same instruction used previously: the model should answer "I don't know" when it cannot find sufficient evidence in the context, even though this option is not explicitly included in the dataset's answer key. Interestingly, the "Understanding the Context" stage achieved better performance compared to the final stage. In this earlier step, the model answered "I don't know" 52 times, whereas in the final step it selected this response 144 times. Despite the variation in performance across stages, CRF² outperformed the results presented in [Ming et al. 2025] on the same counterfactual context dataset, although the authors of that work used the GPT-4o model.

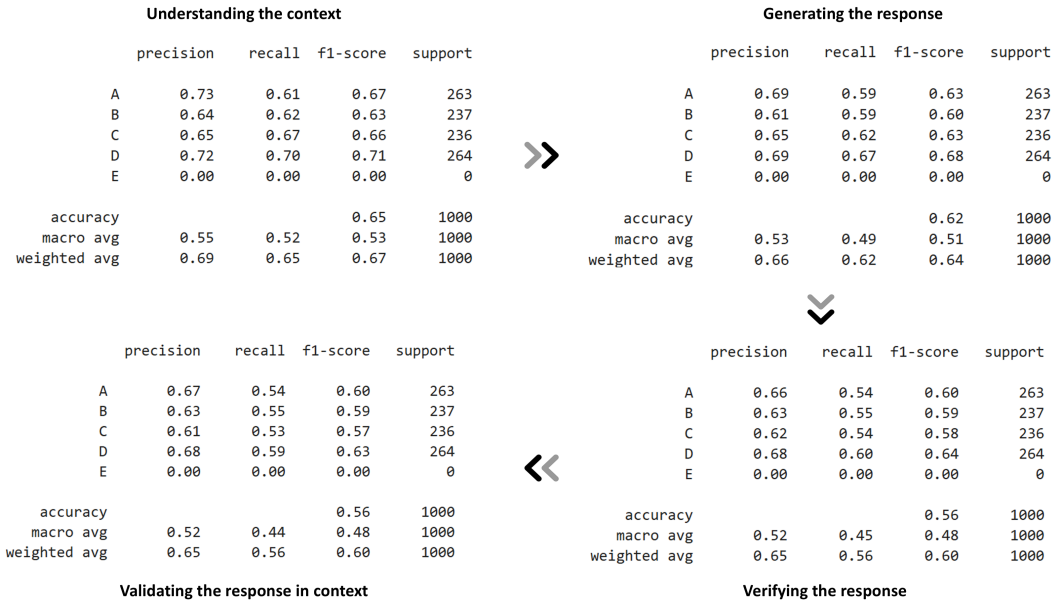

Figure 2. Performance results of the CRF^2 framework at each stage on the FaithEval dataset under the counterfactual context scenario. The tables display detailed evaluation metrics for each item in a set of 1,000 questions.

6. Discussion

In the experiment conducted using the RealTime QA benchmark, it was observed that starting from the combination of UC + AG, the addition of subsequent stages did not lead to performance improvements. This raises two important points for analysis: (i) whether the answer verification stage properly evaluated the responses generated in the previous step, and (ii) whether the context validation stage was able to reinforce the verified answer using specific excerpts from the provided context. The framework disagreed with the official answer key on four questions, which deserve particular attention, as an analysis of the justifications generated by the model suggests that the answer key itself may be incorrect. The full outputs generated by the models, are available in the GitHub repository. The four questions with disagreements are organized in a separate file within the repository.

The results obtained from the experiment using the FaithEval benchmark were lower than those achieved with RealTime QA. One of the main reasons for this difference lies in the distinct nature of each benchmark. In the case of FaithEval, we used questions with counterfactual contexts, i.e., scenarios that include statements contradicting common knowledge or widely accepted facts, which introduces an additional level of complexity compared to RealTime QA. Although the performance surpassed the results reported in [Ming et al. 2025], it is necessary to reassess the proposed framework and make adjustments to improve its effectiveness. As in the previous experiment, the complete outputs are available in the previously mentioned repository.

7. Conclusions and Future Work

We present CRF^2, a framework designed to guide large language models in generating responses that remain faithful to the provided context. The motivation

behind CRF^2 stems from the growing challenge of hallucinations in LLM outputs when models rely excessively on training data without adequately considering the provided context, especially in fact-sensitive scenarios. Our approach achieved accuracy higher than baseline methods, representing a relevant contribution to the field under investigation. As limitations and challenges, we highlight the need for benchmark adaptations to better evaluate fidelity in LLMs and the risk of reasoning breakdown in the final stage due to the search for direct answers. All code and outputs from the experiments have been made publicly available. As future work, we plan to refine the steps, ensuring complementarity between them to achieve better performance, evaluate the framework using other models, such as GPT-4o and DeepSeek, as well as apply additional benchmarks like LogiQA to assess CRF^2's performance on tasks that require complex reasoning.

References

Bao, G., Zhang, H., Wang, C., Yang, L., and Zhang, Y. (2024). How likely do llms with cot mimic human reasoning?

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., and Weston, J. (2023). Chain-of-verification reduces hallucination in large language models.

Huang, G., Long, Y., Luo, C., Shen, J., and Sun, X. (2024). Prompting explicit and implicit knowledge for multi-hop question answering based on human reading process.

Huyen, C. (2025). *AI Engineering: Building Applications with Foundation Models*. O'Reilly.

Jurafsky, D. and Martin, J. H. (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition. Online manuscript released January 12, 2025.

Kasai, J., Sakaguchi, K., Takahashi, Y., Bras, R. L., Asai, A., Yu, X., Radev, D., Smith, N. A., Choi, Y., and Inui, K. (2024). Realtime qa: What's the answer right now?

Li, Y., Zhou, K., Qiao, Q., Nguyen, B., Wang, Q., and Li, Q. (2024). Investigating context-faithfulness in large language models: The roles of memory strength and evidence style.

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., and Gao, J. (2024). Large language models: A survey.

Ming, Y., Purushwalkam, S., Pandit, S., Ke, Z., Nguyen, X.-P., Xiong, C., and Joty, S. (2025). Faitheval: Can your language model stay faithful to context, even if "the moon is made of marshmallows".

OpenAI (2022). Chatgpt: Optimizing language models for dialogue.

Pan, Z., Luo, H., Li, M., and Liu, H. (2024). Chain-of-action: Faithful and multimodal question answering through large language models. *arXiv preprint arXiv:2403.17359*.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., and Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Saravia, E. (2022). Prompt Engineering Guide. *https://github.com/dair-ai/Prompt-Engineering-Guide*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models.

Xiao, T. and Zhu, J. (2025). Foundations of large language models.

Xu, R., Qi, Z., Guo, Z., Wang, C., Wang, H., Zhang, Y., and Xu, W. (2024). Knowledge conflicts for llms: A survey.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models.

Yu, W., Zhang, H., Pan, X., Ma, K., Wang, H., and Yu, D. (2024). Chain-of-note: Enhancing robustness in retrieval-augmented language models.

Zhang, Y., Yang, J., Yuan, Y., and Yao, A. C.-C. (2024). Cumulative reasoning with large language models.

Zhou, W., Zhang, S., Poon, H., and Chen, M. (2023). Context-faithful prompting for large language models. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, Singapore. Association for Computational Linguistics.