CSE 576
TOPICS IN NATURAL LANGUAGE PROCESSING
Project (Final Report)

1. Problem Statement

Hallucination in Large Language Models (LLMs) remains a critical challenge, particularly in complex reasoning tasks. This project aims to explore methods for generating optimal questions to detect hallucinations in LLM-generated text, focusing on legal reasoning scenarios. Five advanced reasoning and verification approaches will be applied to various LLMs, including Llama-3.1-8B-Instruct, Llama-3.2-3B-Instruct and Gemini-1.5-Flash. The evaluation will assess models' ability to correct faulty reasoning, compare question-generation effectiveness, and determine the most robust verification method for minimizing hallucinations and enhancing factual accuracy.

2. Approach to Address the Problem (with implementation details in this report)

- TREACC (Topic, rule, explanation, analysis, counter arguments, conclusion)

Using the TREACC approach introduced in the paper *"Exploring the Effectiveness of Prompt Engineering for Legal Reasoning Tasks"*, the methodology can be implemented with the LLama-3.1-8B-Instruct/Llama3.2-3B-Instruct/ Gemini-1.5-flash model to enhance structured reasoning. The process involves generating structured components: **Topic**, **Rule**, **Explanation**, **Analysis**, **Counter arguments**, and **Conclusion** based on a provided **context**, **questions**, **options**, and **ground truth**.

For each reasoning task, the model first identifies the **Topic**, outlining the central theme or issue in the given context. Next, it derives the **Rule**, summarizing the governing principle, legal doctrine, or logical framework relevant to the problem. The **Explanation** expands the rule by connecting it to the context, clarifying its applicability. In the **Analysis** stage, the model methodically applies the rule to the facts, linking them logically to form a robust argument. **Counter arguments** are then introduced to explore alternative interpretations or objections, strengthening the reliability of the reasoning. Finally, the **Conclusion** provides a definitive answer or decision, justified by the preceding analysis. Then combine all of the components to go through LLM.

- Self-verification approach

The self-verification approach is implemented to evaluate how well the self-verification prompting approach performs across meta-llama/Llama-3.1-8B-Instruct, meta-llama/Llama-3.2-3b and gemini-1.5-flash models and to identify its effectiveness, limitations, or adaptability. It is a two step process. Step 1 required the model to perform **legal reasoning steps and give a solution**. The input consists of legal_context, question, options, analysis(ground_truth). In step 2, where self verification occurs, the output from step 1 as one of the inputs along with the legal_context, question, options, and analysis(ground_truth) the model **generates verification questions and answers** and gives feedback, analyzing to provide a final. Overall, the proposed methodology effectively generates self-verification questions and evaluates them to produce legally accurate, context-sensitive, and logically sound responses, making it a robust solution for tasks involving complex legal reasoning.

- Attention-based Verification Approach

The proposed approach enhances the legal reasoning capabilities of the meta-llama/Llama-3.1-8B-Instruct, meta-llama/Llama-3.2-3b and gemini-1.5-flash model by using an Attention-based Verification with Cross-Step Focus framework, inspired by the InstructGPT methodology. This process is divided into three stages: first, the model generates a step-by-step explanation based on the legal context, question, options, and analysis, ensuring logical consistency. In the second stage, it **identifies critical legal terms and formulates verification questions** to assess the accuracy of its reasoning. Finally, the model **evaluates its reasoning steps, corrects errors, and refines its response**, ensuring alignment with the legal context and reducing the likelihood of hallucinations. This iterative process improves the precision and reliability of the model's output, ensuring contextually accurate and legally sound responses.

- Location Based Error Correction in Reasoning

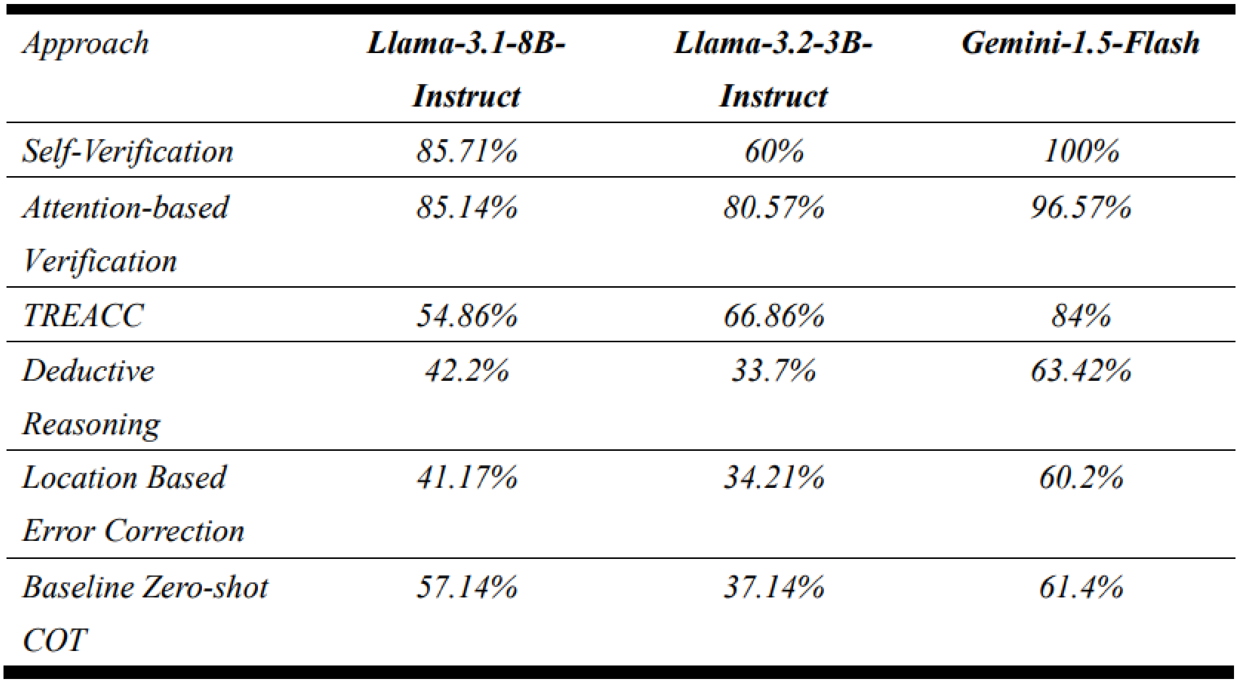
The proposed approach, inspired by the paper *"LLMs Cannot Find Reasoning Errors but Can Correct Them Given the Error Location"*, leverages the LLama-3.1B model to generate reasoning chains for a given legal context. The generated reasoning chain is subsequently evaluated by a verification model designed to **identify faults** or errors in the reasoning steps. Detected errors are transformed into **targeted questions** and mapped to their corresponding **steps in the reasoning chain**. These generated questions are reintroduced into the reasoning chain model, enabling a process of self-verification. This feedback mechanism aims to **pinpoint the exact location of reasoning errors** and guide the model to rectify the faults. By doing so, the model progressively refines its reasoning process, achieving accurate reasoning steps at the end.

- Deductive Verification of CoT Reasoning

The **Natural Program approach**, introduced in "***Deductive Verification of Chain-of-Thought Reasoning***," leverages the LLama-3.1-8B model to generate and **verify multi-step reasoning chains** using structured natural language-based deductive reasoning. Reasoning tasks are **decomposed into discrete steps**, formatted in the Natural Program style, which ensures precision by explicitly **identifying minimal subsets of premises relevant to the context and question**. Each step builds rigorously on prior validated steps, leading to a final answer that **explicitly references the premises** and ensures logical consistency. Verification occurs at every step through an independent subprocess, isolating premises to detect and filter hallucinated or irrelevant reasoning. The method also integrates a **self-verification mechanism** where the model validates the deductive validity of reasoning statements. To enhance reliability, **Unanimity-Plurality Voting** samples multiple reasoning chains, cross-validates them, and selects the consensus. This iterative strategy reduces errors and hallucinations, improving both the accuracy of the final answers and the overall reliability of the reasoning process.

3. Results Obtained from the Approach

The data set consisted of 175 legal reasoning questions, options, and ground truth. Each team member ran the dataset on our respective approaches on 3 different models apart from Zero-shot COT. Our results are tabulated below.



Approach	***Llama-3.1-8B-Instruct***	***Llama-3.2-3B-Instruct***	***Gemini-1.5-Flash***
Self-Verification	*85.71%*	*60%*	*100%*
Attention-based Verification	*85.14%*	*80.57%*	*96.57%*
TREACC	*54.86%*	*66.86%*	*84%*
Deductive Reasoning	*42.2%*	*33.7%*	*63.42%*
Location Based Error Correction	*41.17%*	*34.21%*	*60.2%*
Baseline Zero-shot COT	*57.14%*	*37.14%*	*61.4%*

From the above results we can analyse that Self-verification approach worked better compared to others. Gemini-1.5-Flash model being a much larger dataset produced better results compared to the meta/llama models.

4. Analysis of Results and Findings

1. Performance of the approaches across various models

The five reasoning approaches were evaluated on three LLMs: Llama-3.1-8B-Instruct, Llama-3.2-3B, and Gemini-1.5-Flash. Key findings include:

- Self-Verification: Performed exceptionally well on Gemini-1.5-Flash (100%) but moderately on Llama-3.2-3B (60%). This indicates the approach's reliance on the model's contextual understanding and verification capabilities.
- Attention-based Verification: Performed consistently strong across models, with 85.14% on Llama-3.1-8B and 96.57% on Gemini-1.5-Flash, showing its adaptability and effectiveness in different reasoning environments.
- TREACC: Showed variable performance, peaking at 84% on Gemini-1.5-Flash but only achieving 54.86% on Llama-3.1-8B, suggesting sensitivity to model-specific reasoning styles.
- Deductive Verification: The weakest performer, achieving only 42.28% on Llama-3.1-8B, highlighting challenges in decomposing legal reasoning tasks into deductive chains.

2. Correcting Zero-shot COT Errors

- Models effectively corrected previously failed Zero-shot COT samples using structured reasoning methods.
- **Self-Verification and Attention-based Verification** successfully corrected most faulty samples, especially in Gemini-1.5-Flash.
- There were 2-3 instances where the samples were correctly answered by Zero-shot COT in both the approaches mentioned, but however they were very minimal.
- Many questions answered correctly by Zero-shot COT were incorrectly handled by **Deductive Verification and Location based error correction** approaches due to over-complicated reasoning steps. This highlights the trade-off between structured reasoning depth and inherent model capabilities in simpler tasks.

3. Investigation and Justification of Method Performance

- Self-Verification approach worked well due to its iterative evaluation, enabling contextual refinement, especially when the model's verification prompts are aligned with the task's legal context. However, performance dropped on models with limited verification capabilities, such as Llama-3.2-3B.

- Attention-based Verification approach worked well. Its focus on critical terms and cross-step evaluation made it robust, especially in legally dense scenarios. Its ability to isolate reasoning faults led to consistent high scores.The iterative, multi-stage approach allowed the model to refine reasoning; however, the reevaluation of correct steps led to occasional misclassifications, slightly affecting overall performance.
- TREACC's structured reasoning approach, while thorough, introduced complexity components but no self-verification steps to check by itself leads to hindered its ability to detect errors accurately. Similar with zero-shot CoT but with more given details from contexts. The intricate process led to misinterpretations, which resulted in a lower accuracy.
- CoVE excelled in simpler tasks by leveraging Llama_3.1_8b's pre-trained knowledge, effectively handling straightforward problems without complex option selection. Deductive Verification of CoT Reasoning, with its structured premise-conclusion validation, was more suited for complex tasks requiring rigorous option elimination and context alignment. However, CoT struggled with gaps in logical reasoning rules and difficulty addressing human-centric, non-monotonic conclusions in nuanced legal reasoning.
- The model failed to identify the correct location of the reasoning error while implementing the **Location Based Error Correction** methodology, resulting in the generation of wrong verification questions. This resulted in the reduced overall accuracy score.

4. Empirical Analysis of Approaches on Llama-3.1-8B-Instruct model

Both Subjectively and Objectively, Self-Verification approach outperformed all the other approaches for the Llama-3.1-8B-Instruct model.

Self-Verification stood out for its ability to generate context-sensitive, logically precise questions. Given its 85.71% accuracy in Llama-3.1-8B, it showed the best performance in minimizing error propagation due to its simpler two-step design.

Attention-based Verification performed well overall but had a slightly lower accuracy (84.57%), suggesting that while its multi-stage approach generated effective questions, the process of revisiting reasoning steps sometimes led to the misclassification of correct steps. Despite this, its attention to critical terms helped achieve strong logical consistency.

5. Individual Contributions of Team Members

a. Chao-Shiang, Chen - 20%
   - Implemented the TREACC Approach

b. Ankitha Dongerkerry Pai - 20%
   - Implemented the Self-Verification Approach

c. Rakshita Madhavan - 20%
   - Implemented the Attention-based Verification Approach

d. Suraj Kumar Manylal - 20%
   - Implemented the Location based Error correction Approach

e. Bala Sujith Potineni - 20%
   - Implemented the Deductive Verification Approach

6. References

[1] Yu, Fangyi, et al. “Exploring the Effectiveness of Prompt Engineering for Legal Reasoning Tasks.” *Annual Meeting of the Association for Computational Linguistics*, 1 Jan. 2023, https://doi.org/10.18653/v1/2023.findings-acl.858. Accessed 3 Nov. 2023.

[2] Hong, Ruixin, et al. “A Closer Look at the Self-Verification Abilities of Large Language Models in Logical Reasoning.” *ArXiv (Cornell University)*, 14 Nov. 2023, https://doi.org/10.48550/arxiv.2311.07954. Accessed 15 Apr. 2024.

[3] Zhan, Ling, et al. “Deductive Verification of Chain-of-Thought Reasoning” *ArXiv (Cornell University)*, 6 June 2023, https://arxiv.org/abs/2306.03872. Accessed 3 Oct. 2023.

[4] Tyen, et al “LLMs cannot find reasoning errors, but can correct them given the error location”, Feb 2024, https://aclanthology.org/2024.findings-acl.826

[5] Zhang, Tianhang, et al. "Enhancing uncertainty-based hallucination detection with stronger focus." *arXiv preprint arXiv:2311.13230* (2023).