CSE 576-NLP Project Final Report -Team Lexicoders

1. Problem statement

The objective of this study is to comprehensively evaluate and compare the performance of both open-source and closed-source Large Language Models (LLMs) on logical reasoning tasks. Specifically, the focus will be on evaluating models using benchmark datasets like LogiQA and ARCT, which are designed to test logical reasoning capabilities in natural language processing. The evaluation aims to analyze the reasoning performance of these models, identifying strengths and deficiencies in their reasoning chains and outputs.

A key aspect of this research will be the detailed examination of errors made by the models, with the goal of developing a robust error taxonomy that categorizes the types of reasoning failures encountered. This taxonomy will serve as a foundational resource to provide actionable insights for improving reasoning processes in LLMs. Furthermore, the taxonomy will be leveraged to enhance an auto-evaluator LLM model, which will be designed to assess reasoning quality and guide the refinement of reasoning chains in both training and deployment phases.

Through this study, we aim to advance the understanding of reasoning mechanisms in LLMs and contribute to the development of more effective and reliable language models for tasks requiring logical reasoning.

2. Approach to Address the Problem

2.1 Performance measure

In this study, we selected two Large Language Models (LLMs)-an open-source model (LLaMA-3) and a closed-source model (Gemini)- to evaluate their performance on logical reasoning tasks. The evaluation was conducted using two widely recognized datasets, LogiQA and ARCT, which are specifically designed to benchmark reasoning abilities in natural language processing systems. To ensure consistency across experiments, we employed the zero-shot Chain-of-Thought (CoT) prompting technique, a method that encourages models to generate intermediate reasoning steps before arriving at a final answer.

Initial Results Obtained from the Chosen Approach

Starting of with accuracy of selected models on datasets -

Models	**Llama-3**	**Gemini**
Accuracy (LogiQA)	**31.34%**	**61.14%**

Accuracy (ARCT)	**54.28%**	**85.14%**

Accuracies of LLMS on both datasets

2.2 Manual Analysis and Error Categorization

To gain deeper insights into the models' reasoning capabilities and deficiencies, we conducted a manual analysis of 120 samples from the LogiQA dataset, equally divided into 60 correct and 60 incorrect predictions. Manual(Human) evaluators were provided with step by step guidelines for evaluation. Manual(Human) evaluators divided each reasoning chain into a series of sentences and analysed each sentence and labeled similar errors.

Particular attention was given to the reasoning chains generated by the models, as these chains provide valuable information about the decision-making process. Each labelled error was categorized into one of the following five broad categories based on the alignment of premises and conclusions and 9 sub categories based on the nature of the error:

Broad categories:
RR: Wrong Premise, Wrong conclusion
RR: Wrong Premise, Wrong conclusion
RR: Wrong Premise, Wrong conclusion
RR: Wrong Premise, Wrong conclusion
NC: No Conclusion: Sentences containing only information from Problem, options or premises from previous steps fall under this category.

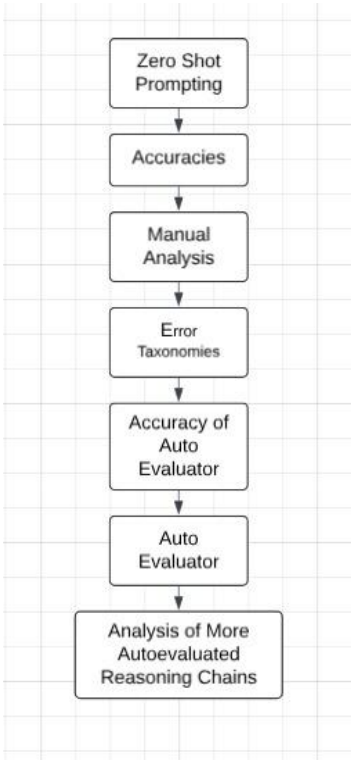
Sub categories:
Wrong premise:

1. Error propagation: Premise taken from previous wrong conclusion.
2. Misinterpretation: Misinterpretation of passage, options or previously concluded steps in premise.

Wrong conclusion:

1. Insufficient information: Draws conclusions based on insufficient information and misses important information such as passage conditions.
2. Wrong assumption: Assumes conditions which leads to wrong conclusion.
3. Contradiction: Draws Conclusion that contradicts a previous premise
4. Misinterpretation: Misinterpretation of passage, options or previously concluded steps in premise[prominent for passage based problems].

5. Hallucination: Considers out of context information leading to wrong conclusion
6. Wrong Reasoning: Gives wrong conclusion irrespective of premise[prominent for clue and arrangement based problems].
7. Error propagation: Conclusion drawn from wrong premise.



Flow of the project

2.3 Creation of the Custom GPT-Based Auto Evaluator

Using the insights gained from manual evaluation, we developed a custom GPT-based Auto Evaluator to automate the process of reasoning chain analysis and annotation. The Auto Evaluator was trained and configured using the following inputs:

1. **Manual Evaluation Instructions:** Detailed guidelines provided to human evaluators for categorizing reasoning chains.
2. **Analysis Instructions:** Clear directives for the model to follow during automated analysis and classification.
3. **Error Taxonomy:** The comprehensive set of categories and subcategories developed during the manual analysis phase.
4. **Annotated Examples:** Manually evaluated reasoning chains used as examples to guide the Auto Evaluator.

The Auto Evaluator was designed to process a variety of inputs, including the passage, question, answer options, LLM generated reasoning chain, predicted answer, and the correct answer. Using these inputs, the model was tasked with replicating the manual evaluation process and classifying reasoning chains based on the error taxonomy and a set of instructions. The prompt given to the custom gpt was as follows:

{ {Instruction}:{...}

{"Passage"}: {...}

{"Question"}: {...}

{"Options"}: {...}

{"Reasoning Chain"}: {...}

{"Correct Answer"}: {...}

{"Predicted Answer"}: {...} }

The instructions outline a framework for error analysis by breaking reasoning chains into individual premises and conclusions. Sentences starting with words like "so" or "therefore" use the previous conclusion as their premise. Each sentence is compared to the correct solution and classified into main categories (WW, WR, RR, RW, NC) based on premise and conclusion accuracy. Incorrect premises or conclusions are further categorized into subcategories, and classifications are reviewed for accuracy.

3. Results Obtained from the Chosen Approach

3.1 Evaluation of the Auto Evaluator's Performance

After the development of the custom GPT-based Auto Evaluator, its performance was rigorously tested to ensure that it could accurately replicate the manual evaluation process. The goal was to validate its ability to classify reasoning chains generated by LLMs into the predefined error taxonomy, both at a broad and a granular level.

To achieve this, we compared the Auto Evaluator's output against a set of manually analyzed reasoning chains. These manually evaluated chains served as the ground truth, representing human judgments of the reasoning quality. The evaluation focused on two levels of accuracy:

1. Broad Category Accuracy: The ability of the Auto Evaluator to classify reasoning chains into the five main categories:

- NC, RR, RW, WR, WW

2. Subcategory Accuracy: The ability of the Auto Evaluator to classify reasoning chains into more detailed subcategories that reflect nuanced errors or reasoning patterns.

3.2 The results of this comparison revealed the following:

- Broad Category Accuracy: The Auto Evaluator achieved an impressive accuracy of 81.98% when classifying reasoning chains into broad categories.
- Subcategory Accuracy: At the more granular subcategory level, the Auto Evaluator achieved an accuracy of 56.79%.

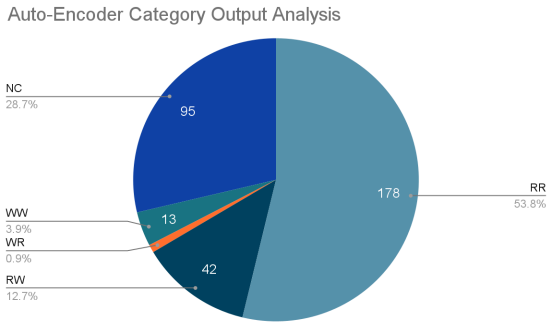
	Total	Matched
Category	311	255
Sub-Category	81	46

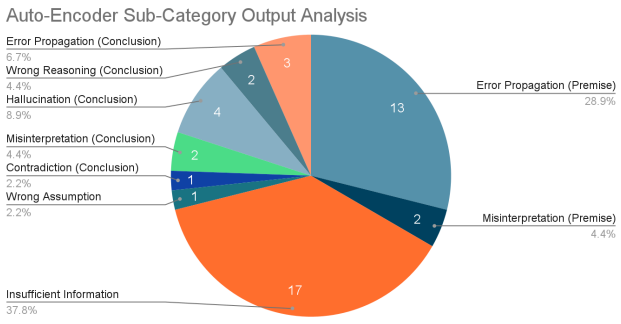
Result of Comparison of manually analyzed chains and auto evaluated chains

These results indicate that the Auto Evaluator performs well in capturing general trends in reasoning quality but faces challenges when analyzing more intricate and subtle aspects of reasoning errors.

4. Analysis of Results and Findings (~1 pages)

To understand the types of errors LLMs make, we conducted an in-depth analysis of 30 reasoning chains evaluated by the auto-evaluator. Our findings reveal that the majority of individual sentences were categorized as RR (Right Premise, Right Conclusion), indicating that LLMs often reason correctly at a granular level. This was followed by a significant proportion of sentences classified as NC (No Conclusion), suggesting that incomplete reasoning processes are another frequent outcome.





When errors occurred, they were predominantly caused by Error Propagation(28.9%), where an incorrect conclusion from a previous step was used as a premise for subsequent reasoning. This cascading effect highlights the challenge of maintaining logical consistency across multi-step reasoning tasks. Additionally, errors due to Misinterpretation of Premises(37.8%) were also observed, where LLMs misrepresented or misunderstood the given information. These findings underscore key areas for improvement in LLM reasoning, particularly in handling complex reasoning chains and ensuring accurate premise interpretation.

5. Individual Contributions of Team Members (~0.5 page)

Team members	Contribution in %
Atharva Chundurwar	20%
Sai Teja Bandaru	20%
Bhaskar Bose	20%
Dhruv Rakeshkumar Prajapati	20%
Tejas Ajay Parse	20%

6. References

1. **Habernal, I., Wachsmuth, H., Gurevych, I., & Stein, B. (2017). The argument reasoning comprehension task: Identification and reconstruction of implicit warrants.** ***arXiv preprint arXiv:1708.01425*****.**
2. **Tyagi, N., Parmar, M., Kulkarni, M., Rrv, A., Patel, N., Nakamura, M., ... & Baral, C. (2024). Step-by-Step Reasoning to Solve Grid Puzzles: Where do LLMs Falter?.** ***arXiv preprint arXiv:2407.14790*****.**
3. **Liu, J., Cui, L., Liu, H., Huang, D., Wang, Y., & Zhang, Y. (2020). Logiqa: A challenge dataset for machine reading comprehension with logical reasoning.** ***arXiv preprint arXiv:2007.08124*****.**