Can LLM's Solve Brain Teasers?

Alex Gagnon
(4537656)
School of Electrical Engineering and Computer Science
University of Ottawa

John Joseph Zagariah Daniel
(300382174)
School of Electrical Engineering and Computer Science
University of Ottawa

Kartik Banga
(300344046)
School of Electrical Engineering and Computer Science
University of Ottawa

I. Introduction

In today's tech landscape, Natural Language Processing (NLP) has emerged as a crucial field with widespread applications. Tech giants like OpenAI, Google, Meta, and Microsoft are continuously releasing Large Language Models (LLMs) to meet the rising demand and dominate the LLM market. While these models excel in many areas, they often struggle with tasks requiring unconventional or lateral thinking.

Lateral thinking, a form of thinking that involves solving problems through an indirect and creative approach, presents a challenge for LLMs. Brain teasers, which often require lateral thinking to solve, can serve as a testbed to evaluate the ability of LLMs to engage in such thinking. For example, how can a cowboy riding into town on Friday and leaving on Wednesday, but only spend two days in town? The answer, "Friday and Wednesday are the names of two horses", has nothing to do with the context of the riddle, and challenges implicit premises and requires thinking beyond rationality and logical rules.

II. Problem Statement

This project aims to assess the performance of recent LLMs, Google's Gemma and Meta's Llama 3, in solving brain teasers using a multiple-choice question-and-answer dataset from the SemEval 2024 Task 9 competition. To solve this problem, we considered three potential strategies:

1. Fine-Tuning for lateral-thinking-based MCQs

Various transformer models like BERT, ALBERT, Roberta, XLNet, etc. support AutoModelForMultipleChoice, which means they are pre-trained for MCQ's and hence are able to understand their structure and the desired output. For training, pairs of questions or options were created for each available option, they were preprocessed with tokenization, the sentences were padded to a specific length, and the model fine-tuned for several epochs to answer brain teasers specifically. However, many researchers, like Jiang et al. (2023c), have claimed that fine-tuning the model on brainteasers or tasks requiring out-of-box thinking doesn't really improve the performance of small models like BERT.

2. Maximum Likelihood with Autoregressive models

Similar to the previous approach, the MCQ is split into 4 question/option pairs, but then an autoregressive model that supports AutoModelForCausalLM like GPT-2 is used to compute the likelihood that the model would have generated the option given the question. The option with the highest likelihood is the model's choice. For this task, no fine-tuning is required, and we assume that these larger autoregressive LLMs that are trained on more data and with greater numbers of parameters would be better able to perform lateral thinking.

3. Prompt Engineering

However, recent research into prompt engineering suggests that simply crafting prompts in a certain fashion can lead to high quality results without requiring fine-tuning at all. It also allows for easily adding necessary context to try to maximize the likelihood of obtaining the desired output. As such, the remainder of this paper focuses solely on using prompt engineering strategies to solve brain teasers.

By identifying which prompts perform best on this task, we can gain insight into the capabilities and limitations of various models in handling lateral thinking challenges, which has implications for their use in real-world applications requiring creative problem-solving.

III. Related Works

The field of reasoning in natural language processing is experiencing significant advancements, particularly through the development of pre-trained language models, such as large language models (LLMs). These advancements cover various domains of reasoning including commonsense, mathematical, and logical reasoning, supported by benchmarks like BigBench and datasets such as CommonsenseQA, WinoGrande, and RiddleSense.

Significant research has focused on enhancing the capabilities of these models in commonsense reasoning. Nie et al. (2020) [1] and Trinh and Le (2019) [2] have explored the use of unsupervised methods and vast text corpora to train models that perform competitively without relying on annotated databases. The architectural prowess of transformer models such as BERT and GPT has been pivotal in elevating performance in natural language understanding and complex question-answering tasks, harnessing massive text corpora to discern plausible answers from multiple options.

Prompt engineering has emerged as a crucial technique for ensuring the LLM provides meaningful output. Whether the response from a chatbot is correct, concise, and useful depends greatly on the quality of the prompt. Techniques such as contrastive explanations, where models generate detailed explanations that delineate differences between answer choices, have shown to improve both performance and the relevancy of explanations (Paranjape et al., 2021) [3]. Similarly, Liu et al. (2022) [4] have developed methods for 'generated knowledge prompting', which enhances LLMs' responses by integrating generated knowledge statements directly into prompts.

Another strategy to enhance reasoning involves the integration of unstructured knowledge resources such as Wikipedia, into the prompts. By incorporating these external knowledge sources directly into the input, models gain access to a broader context that enhances their ability to perform reasoning tasks. (Yongfeng Huang., 2023) [5]. Advanced methods in prompt engineering also explore generating diverse opinions and engaging in complex reasoning tasks, highlighting the adaptability and creativity possible with LLMs (Xavier Amatriain.,) [6]. Additionally, studies have explored the efficacy of automatically discovered chain-of-thought prompts, which adapt across different models and datasets to optimize reasoning performance, though they note challenges such as the limitations of dataset subsampling and model variability (Konstantin Hebenstreit.,) [7].

IV. Dataset

The BrainTeaser dataset consists of multiple-choice questions where each question has four options: a correct answer and 3 distractors. The dataset is divided into two parts: Sentence Puzzle (627 samples) and Word Puzzle (492 samples), each of which requires some lateral thinking.

1. Sentence Puzzle: Brain teasers that challenge participating entities to interpret sentence snippets in unexpected ways. Example: A man shaves every day, yet keeps his beard long. A) He is a barber. B) He wants to maintain his appearance. C) He wants his girlfriend to buy him a razor. D) None of the above. (Answer: A)

2. Word Puzzle: Brain teasers that challenge participating entities to rethink word meanings, focusing on letter composition in the primary question. Example: What part of London is in France? A) The letter N. B) The letter O. C) The letter L. D) None of the above. (Answer: A).

During preprocessing, additional steps were needed due to formatting issues. For example, several entries in the dataset had values that didn't match the published schema (i.e. the label was a float instead of a string), so needed to be normalized accordingly.

The dataset does not have a dedicated evaluation partition that contains labels, so 10% of the training set was used. Then the training dataset was further split 80/20 for training/validation sets. We only used the training and validation sets during our initial analysis of using BERT with AutoModelForMultipleChoice.

The remainder of this paper focuses exclusively on the evaluation set, since the Prompt Engineering approach is essentially zero-shot learning and no fine-tuning was required.

V. Methodology

One method for determining whether LLMs have lateral thinking abilities is to formulate the riddle as a chat prompt and have the LLM generate the answer. A key strategy is to give the model context about the task, indicating that it must participate in unconventional thinking. To determine the impact of different prompt strategies, we designed six chat templates (three manually crafted, and three by asking modern LLMs to output a prompt).

(I) Basic approach, which provides no special detail, which means the prompt doesn't contain any information about lateral thinking.

```
    'manual-basic-mcq': Template('''
Which of the following options answers this multiple-choice question: $question
A) $a
B) $b
C) $c
D) $d
Provide the correct option as A, B, C, or D.
Answer: '''),
```

Figure 1: Prompt Engineering: Basic MCQ Template

(II) Riddle Context, which provides additional information to the LLM that the question may require lateral thinking;

```
    'manual-basic-riddle': Template('''
Which of the following options answers this multiple-choice riddle: $question
A) $a
B) $b
C) $c
D) $d
Provide the correct option as A, B, C, or D.
This is a riddle so the answer may involve word play or lateral thinking.
Answer: '''),
```

Figure 2: Prompt Engineering: Basic Riddle Template

(III) Chain of Thought, in which the LLM is given a hint that the question may require lateral thinking and is asked to explain the output, step by step.

```
    'manual-chain-of-thought': Template('''
Which of the following options answers this multiple-choice riddle: $question
A) $a
B) $b
C) $c
D) $d
Provide the correct option as A, B, C, or D.
This is a riddle so the answer may involve word play or lateral thinking.
Explain how you deduce the answer step by step.
Answer: '''),
```

Figure 3: Prompt Engineering: Chain of Thought Template

These prompts were carefully designed after conducting small initial experiments and examining the output to identify which styles of prompting led to correct and easily parsable responses.

The other three prompts were created by asking different LLMs (Gemini, Llama 3, and ChatGPT4) to generate them by prompting them with the question “Can you generate a prompt template for me that is designed to help an LLM answer a multiple choice riddle?”

```
    'llama-3-prompt-generation': Template('''
Riddle: $question
Options:
A) $a
B) $b
C) $c
D) $d
Choose the correct answer:
(Select one of the above options by typing A, B, C, or D)'''),
```

Figure 4: Prompt Engineering: Llama-3-prompt Template

Two modern LLMs were used for comparison, Google’s Gemma (7b-it), and Meta’s Llama 3 (8b-it). Both are relatively large models that can fit into consumer grade hardware (e.g. Google Colab environment with T4 GPU, offering 15Gb of RAM) using memory optimizing techniques such as reduced precision parameters and offloading unused models from the GPU when not being used.

The HuggingFace set of python libraries was used to load the models and dataset. For each model and entry in the evaluation datasets, the prompt template was interpolated with the question and options, and then used by the LLM to generate the response. Sampling was used with top_k = 50 and top_p = 0.95 as hyperparameters, to ensure that several potential responses were considered instead of greedily only taking the next most likely token at each phase.

The responses were stored to a log file and then manually reviewed to extract the answer. A heuristic algorithm for automatically identifying the LLMs choice worked for most cases, however due to the inherent “chatiness” of the responses (where each answer would follow a different structure), manual intervention was required to make sure the results were recorded correctly. The models and prompting strategies are then compared based on the accuracy of the responses.

Some answers were ignored from the results such as when the LLM would refuse to answer due to their safety guardrails or because they hadn't yet output a guess before the max token limit. For example, given the following sentence puzzle excerpt: "Although Dorothy has never stated suicide ideas, whenever she feels down or depressed, she makes plans to go to a very high location that she has frequented before. Once there, she makes the decision to jump from the enormous height. The amazing thing is that she has never experienced an injury as a result of her leap, and she even claims to feel much better afterwards. What is going on here, and why has she never been hurt or, worse still, killed as a result of her dangerous behavior, if she isn't suicidal", the model would refuse to answer. However, when the same puzzle was phrased differently as "Dorothy has never expressed suicidal thoughts, but whenever she experiences feelings of sadness or depression, she makes arrangements to travel to a very high spot that she has visited on numerous occasions. Once there, she proceeds to jump from that great height. The fascinating fact though is she has never been injured from this leap, and in fact, she tells everyone that she feels much better afterward" the LLM correctly guesses that Dorothy is a skydiver.

Additionally, two of the configurations did not run to completion due to resource limits when using Google Colab. However, since the only measure considered is accuracy and the number of samples was still relatively large, we included them in the results.

The full code is available at https://github.com/alexgagnon/CSI5386-Final

VI. Experiments And Results

Table 1 shows the results of prompt engineering for sentence puzzles with Llama 3 and Table 2 shows the results of prompt engineering for word puzzles with Llama 3.

A. Results of Prompt Engineering with llama-3-8b-it

Prompt Engineering Strategy and Accuracy for Sentence Puzzles - Llama3

Basic	0.5760
Riddle Context	0.6847
Chain of Thought	0.6086
Gemini	0.6538
Llama 3	0.7065
Chat GPT	0.7065

Table 1: Prompt Engineering Strategy and Accuracy for Sentence Puzzles - Llama3

From the table, we can see that for sentence puzzles, Llama 3 gives the best performance with the Chat-gpt4 template (accuracy = 0.706).

Prompt Engineering Strategy and Accuracy for Word Puzzles - Llama3

Basic	0.4035
Riddle Context	0.5000
Chain of Thoughts	0.6527
Gemini	0.3888
Llama 3	0.5833
Chat GPT	0.5694

Table 2: Prompt Engineering Strategy and Accuracy for Word Puzzles - Llama 3

For word puzzles, Llama 3 gives the best performance with the Chain of Thought template (accuracy = 0.652), but with other templates, we recorded a significant drop in accuracy.

B. Results of Prompt Engineering with gemma-7b-it

Prompt Engineering Strategy and Accuracy for Sentence Puzzles - Gemma

Basic	0.2608
Riddle Context	0.2500
Chain of Thought	0.2826
Gemini	0.3369
Llama 3	0.2717
Chat GPT	0.2608

Table 3: Prompt Engineering Strategy and Accuracy for Sentence Puzzles - Gemma

From the table, we can see that for sentence puzzles, Gemma gives the best performance with the Gemini template (accuracy = 0.336).

Prompt Engineering Strategy and Accuracy for Word Puzzles - Llama 3

Basic	0.3472
Riddle Context	0.3611
Chain of Thought	0.3750
Gemini	0.4027
Llama 3	0.3750
Chat GPT	0.2777

Table 4: Prompt Engineering Strategy and Accuracy for Word Puzzles - Gemma 3

For word puzzles, Gemma gives the best performance with the Gemini template (accuracy = 0.402).

VII. Discussion

Overall, Llama 3 significantly outperformed Gemma on every configuration. This could be due to many factors, such as the training data for Llama 3 having a higher proportion of lateral thinking puzzle examples, having an architecture that allows for more creatively exploring different samples, or the slightly larger size (8b parameters vs. 7b for Gemma). Llama 3 is also much more recent, however the recency is likely not a major contributor as puzzles and word games have been available to use in training corpus previously.

We note that the Chain of Thought approach seems to work well for Word puzzles, while for sentence puzzles the prompt generated by the same family of models was the top performer (i.e. Gemini for Gemma, and Llama 3 for Llama 3). While Chain of Thought is known to be effective when the solution can be achieved through step by step deduction, why it appears to work so well for Word puzzles specifically is an opportunity for further research.

Anecdotally, there were some interesting findings worth noting. As mentioned previously, even minute changes to the question can lead to drastic changes in the response, ranging from the extreme of the LLM refusing to answer, to where it would be able to correctly answer one but not the other. This concisely demonstrates the impact the structure and makeup of the prompt have on the quality of the response.

LLMs also seem to struggle when the riddle contains mathematical concepts, since it tries to logically solve those aspects of the question. For example, given the question “Which of the following options answers this multiple-choice riddle: What's 3/7 chicken, 2/3 cat, and 2/4 goat?” (Answer: Chicago, Chi + ca + go), the LLM tries to perform mathematical operations and begins to make false claims by the end of the response: “[...]We have three fractions: 3/7, 2/3, and 2/4. Each fraction represents a part of a word. 2. Convert each fraction to a decimal: 3/7 = 0.428571 (approximately) * 2/3 = 0.666666 (exactly) * 2/4 = 0.5 (exactly) 3. Now, combine these decimals to form a new word: Take the first letter of each word, corresponding to the decimal values: + 0.4 (chicken) = C + 0.6 (cat) = C + 0.5 (goat) = G. The resulting word is: C-C-G The answer is a place name that fits this pattern. After some research, I found that: A) Texas is the correct answer! Here's why: * Texas has a "C" in the word, matching the 0.4 (chicken) part. Texas has a "C" in the word, matching the 0.6 (cat) part. Texas has a "T" in the word, which is close enough to the "G" in goat.” Without anthropomorphizing the LLM, it seems to focus intently on the parts of the question where logical reasoning might play a role and ignores the lateral thinking component.

Another interesting finding is that select riddles seem to cause the LLM to ignore ground-truths, even ones that it acknowledges in its response. For example, given the question: “Even if they are starving, natives living in the Arctic will never eat a penguin's egg. Why not?”, the response is “C) Penguins and their eggs are not part of their traditional or cultural diet. Natives living in the Arctic are indigenous peoples such as the Inuit, Yupik, and Aleut, who have traditionally lived in the Arctic regions of North America, Greenland, and Eurasia. Penguins, on the other hand, are found in the Southern Hemisphere, primarily in Antarctica and the surrounding islands, as well as in the southern parts of South America, Africa, and Australia. Therefore, it is unlikely that penguin eggs would be part of the traditional diet of Arctic natives…”. It correctly understands that Penguins are not found in the Artic, but never selects that as the answer, even though it is the most simple explanation.

VIII. Conclusion

Gemma , known for its efficiency and focus on responsible AI, is well-suited for concise prompts.While Llama 3, designed for cutting-edge performance, unlocks the potential for more intricate prompt engineering strategies.Its superior capabilities allows to craft longer, more detailed prompts that provide richer context. Our findings suggest that certain LLMs are in fact able to solve lateral thinking puzzles such as brainteasers, to a relatively high degree of accuracy (for Llama 3, 70% and 65% for Sentence and Word puzzles, respectively). Gemma on the other hand hardly scored above random chance, so there is clear indication that factors like the size or the specific training data used in Llama 3 makes it the superior model for riddles. For future work, additional prompting strategies could be investigated, for example Retrieval Augmented Generation where an accessory database of documents with additional context is queried to improve the prompt and/or the response.

References

[1] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020.Adversarial NLI: A new benchmark for natural language understanding.In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages

4885–4901, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.441

[2] Trieu H. Trinh and Quoc V. Le. 2019.A simple method for commonsense reasoning.http://arxiv.org/abs/1806.02847

[3] Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021.Prompting contrastive explanations for commonsense reasoning tasks.In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4179–4192, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.366

[4] Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019.Bert with history answer embedding for conversational question answering. https://doi.org/10.48550/arXiv.1905.05412

[5] Yongfeng Huang, Yanyang Li, Yicong Xu, Lin Zhang, Ruyi Gan, Jiaxing Zhang, Liwei Wang. MVP-Tuning: Multi-View Knowledge Retrieval with Prompt Tuning for Commonsense Reasoning. https://aclanthology.org/2023.acl-long.750.pdf

[6] Xavier Amatriain. Prompt Design and Engineering: Introduction and Advanced Methods. https://ar5iv.labs.arxiv.org/html/2401.14423v3

[7] Konstantin Hebenstreit, Robert Praas, Louis P Kiesewetter, Matthias Samwald. An automatically discovered chain-of-thought prompt generalizes to novel models and datasets. https://ar5iv.labs.arxiv.org/html/2305.02897

[8] Yifan Jiang, Filip Ilievski1, Kaixin Ma, Zhivar Sourati. BRAINTEASER: Lateral Thinking Puzzles for Large Language Models. https://arxiv.org/pdf/2310.05057.