RiddleMasters at SemEval-2024 Task 9: Comparing Instruction Fine-tuning with Zero-Shot Approaches

Kejsi Take
New York University
Brooklyn, NY
kejsitake@nyu.edu

Chau Tran
New York University
Brooklyn, NY
chau.tran@nyu.edu

Abstract

This paper describes our contribution to SemEval 2023 Task 9: Brainteaser. We compared multiple zero-shot approaches using GPT-4, the state of the art model with Mistral-7B, a much smaller open-source LLM. While GPT-4 remains a clear winner in all the zero-shot approaches, we show that fine-tuning Mistral-7B can achieve comparable, even though marginally lower results.

1 Introduction

Traditionally, the natural language processing (NLP) community has focused on tasks that require objective and complex reasoning. On the other hand, puzzles that defy traditional ways of reasoning have been less explored. Brainteaser, a task at SemEval 2024 (Jiang et al., 2024), aims to fill this gap by investigating the abilities of large language models (LLMs) in more abstract and creative thinking. This competition consists of two sub-tasks: sentence puzzle (SP) and word puzzle (WP). According to the task description , sentence puzzles are brainteasers where the entire sentence snippet defies common sense. Similarly, word puzzles are puzzles where the answer violates the default meaning of the word and focus on the letter composition of the question.

In this work, we investigate a set of zero-shot approaches and compare them with a fine-tuned version of Mistral-7B, an open source 7 billion LLM (Jiang et al., 2023a). For the zero-shot approaches, we compare Mistral-7B with GPT-4, the state of the art transformer model, across various prompts. We find that one-shot approaches using GPT-4 produces the best results across both our tasks. However, tweaking the prompts results in significant accuracy increases for Mistral-7B. We also find that fine-tuned Mistral-7B is the second best model in the sentence puzzle sub-task, indicating that instruction fine-tuning may be a way to get better results with smaller models.

2 Background

The NLP task most related to this competition is question answering (QA), as all riddles consists of a question and multiple potential answers. Question answering has been the focus of extensive prior work (Soares and Parreiras, 2020). Typically, question answering systems consist of three main components: (1) *question processing*, (2) *document processing*, and (3) *answer processing* (Bhoir and Potey, 2014; Soares and Parreiras, 2020). The main goal of the question processing is to extract the keywords from the query so they can be parsed to the document processing component, as well as to identify the type of answer that we need to return (Parsing, 2009). The goal for the document processing system is information retrieval (IR), based on the keywords collected from the previous component. Typically, the IR system's job is to identify a subset of documents relevant to the keywords identified previously (Malik et al., 2013; Gupta and Gupta, 2012).

As the desired output needs to be accurate and succinct, the IR system needs to further break down the relevant documents into smaller units such as passages, paragraphs, or sentences. The final stage of question answering is answer processing, which involves formulating the desired answer based on the knowledge previously retrieved, using a process called span labeling (Parsing, 2009): given a passage, identifying the span of text that can be used to answer the question. These components are largely suitable for answering questions in a way that utilizes straightforward information processing and logical thinking, but struggle against question answering tasks that require creative responses or common-sense reasoning, such as solving puzzles and brainteasers (Jiang et al., 2021, 2023b).

With the recent breakthroughs of LLMs such as BERT (Devlin et al., 2019) or GPT-3 (Brown et al., 2020), we have seen exceptional capabili-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1391–1396
June 20-21, 2024 ©2024 Association for Computational Linguistics

ties of these language models in solving QA tasks, as well as their exhibition of complex reasoning abilities (Hu et al., 2024; Creswell et al., 2022). However, when it comes to creative thinking and common sense reasoning, large language models achieve limited results (Ding et al., 2023; Zhou et al., 2020; AlKhamissi et al., 2022). As such, (Jiang et al., 2023b) generate a dataset of brainteasers to benchmark the performance pf state-of-the-art LLMs in answering puzzles and brainteasers, as a way to test their lateral thinking capabilities. Our work aims to contribute to this domain, by evaluating the performance of multiple zero-shot approaches and our version of fine-tuned Mistral-7B language model (Jiang et al., 2023a) on the same dataset (Jiang et al., 2023b).

Dataset. The authors generated the initial Brainteaser dataset by crawling the puzzles from the internet. However, recent work has shown that memorization is a common problem with LLMs (Carlini et al., 2022). To evaluate lateral thinking instead of memorization, the authors used two reconstruction strategies (semantic and context) to create variants of each puzzle. Semantic reconstruction rephrases the original question and was created via an open-source rephrasing tool (Jiang et al., 2023b). In contrast, context reconstruction was achieved through a combination of GPT-4 prompts and human annotators (Jiang et al., 2023b).

3 System Overview

In this section, we first describe the train and test datasets. Later, we describe our proposed approaches, detailing the prompts used in the zero-shot approaches and the fine-tuning methodology.

3.1 Dataset Description

As mentioned above, we used the provided dataset (Jiang et al., 2023b) which consists of 1,119 data samples, including its reconstruction variants. The questions were divided into two sub-tasks, Sentence Puzzle and Word Puzzle, and further distributed into more than 80 different areas/topics. For more details about the dataset distribution, please refer to (Jiang et al., 2023b).

Train-test split. The data provided by the organizers was split 80:20 between the train and test set. In general there are more examples of sentence puzzles (627 total) than word puzzles (492 total).

Label Distribution. To investigate model bias, we investigated the distribution of labels. As shown in Figure 1, the correct answers are not evenly distributed among all options. In fact, the 4^{th} label (i.e., "None of the above") is the minority label. This label is particularly rare in the train set for word puzzles (9/397) and does not occur in the test set for the same task.

	Sentence Puzzle		Word Puzzle	
train	507	80.8%	396	80.4%
test	120	19.1%	96	19.5%
total	627		492	

Table 1: Number of samples in test and train data.

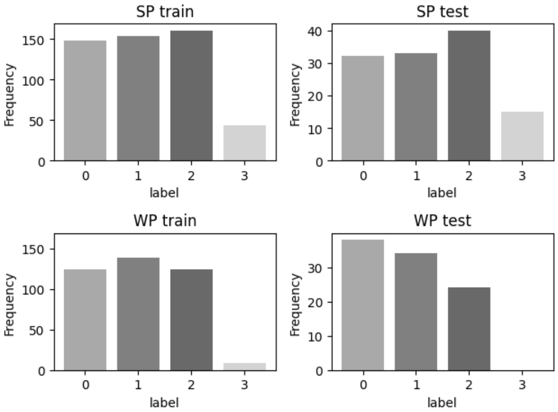

Figure 1: Distribution of the labels between answer choices (answer choices are encoded as 1, 2, 3, 4)

3.2 Zero-Shot Prompting

Given that prior work found that common sense models are not more effective than zero-shot approaches (Jiang et al., 2023b), in this paper we focused only on the latter. We started our evaluation by experimenting with zero-shot solutions. We wanted to compare Mistral-7B with GPT-4, the state of the art transformer model. To provide a thorough evaluation, we experimented with three prompting strategies, which we include in Appendix A. The first one is identical to the one provided by the competition organizers. To formulate the second prompt we leveraged the fact that all riddles contain a "None of the above." answer. This way, the second prompt provides three answer options (excluding "None of the above."). If none of the three answers is returned, we consider that to be "None of the above." In the third prompt, we tried to guide the model to consider all answer options, emulating an approach similar to zero-shot Chain-of-thought (CoT) prompting (Kojima et al., 2022). Similar to the second prompt, here we limit

			Instance-based			Group-based		
Category		**Model**	**Original**	**Semantic**	**Context**	**Ori & Sem**	**All**	**Overall**
Sentence Puzzle								
Random	**-**	**-**	0.175	0.150	0.175	0.000	0.000	0.167
Zero-Shot	**P1**	**GPT-4**	0.825	0.700	0.725	0.675	0.600	0.750
		Mistral	0.275	0.250	0.225	0.225	0.125	0.250
	P2	**GPT-4**	0.850	**0.800**	0.700	0.800	0.625	0.783
		Mistral	0.500	0.500	0.350	0.425	0.250	0.450
	P3*	**GPT-4**	**0.925**	0.750	0.775	**0.750**	**0.675**	**0.817**
Finetuning		**Mistral**	0.800	0.775	**0.800**	0.725	0.650	0.792
Word Puzzle								
Random	**-**	**-**	0.094	0.250	0.219	0.031	0.031	0.188
Zero-Shot	**P1**	**GPT-4**	0.625	0.531	0.656	0.438	0.312	0.604
		Mistral	0.031	0.062	0.094	0.031	0.031	0.062
	P2	**GPT-4**	**0.906**	**0.906**	**0.906**	**0.875**	**0.781**	**0.906**
		Mistral	0.594	0.656	0.625	0.469	0.312	0.625
	P3*	**GPT-4**	0.875	0.906	0.812	0.844	0.719	0.865
Finetuning		**Mistral**	0.844	0.844	0.844	0.781	0.656	0.844

Table 2: Results for the two BRAINTEASER subtasks across all models, prompts (P1, P2, P3) and metrics. Ori = Original, Sem = Semantic, All = Original + Semantic + Context. The best performance among all models is in bold. The random is answer assigned by random choice where the four options have equal probability to be selected. For prompt 3, Mistral-7B did not generate any meaningful responses and therefore we do not include it in this evaluation.

the choices to the three options. Further, we prompt the model to respond "None" if none of the three options is not the answer.

3.3 Instruction-based Fine-tuning

We fine-tuned a sharded version of Mistral-7B[1]. Mistral-7B is an LLM with 7.3 billion parameters. Mistral-7B uses grouped-query attention for faster inference and sliding-window attention to handle longer sequence (Jiang et al., 2023a). We used instruction based fine-tuning, a type of fine-tuning where instructions are used to define downstream tasks. In our case, the instruction was formed by the question and the sample answers.

We fine-tuned the model using Google Colab and used Low-Rank Adaptation (LoRA) (Hu et al., 2021) to make fine-tuning more efficient. LoRA freezes the pre-trained model weights and using rank decomposition matrices into each layer to reduce the number of trainable parameters.

Training Parameters. When fine-tuning with LoRA, one of the parameters is a list of specific layers in the model architecture that will undergo decomposition. While limiting only to attention layers may reduce training time, we targeted all linear layers, as prior work (Dettmers et al., 2024) suggests that this might provide better results. The other significant LoRA parameter is r, the rank of matrices updated during adaptation. However, it has been shown that the value of r does not improve adaptation quality between a certain point[2] and therefore we keep $r = 8$. These approaches result in 21M trainable parameters (0.29%) instead of a total of 7B.

4 Results

The results for all the experiments are included in Table 2. In this section, we discuss and compare all the approaches.

4.1 Zero-Shot Prompting

In the zero-shot experiments, Mistral-7B generally performs worse than GPT-4. This is not surprising, as it is a much smaller model (7 billion parameters vs 1.76 trillion). Further, zero-shot approaches with prompt 2 and prompt 3 perform better than the one with prompt 1. These approaches are also the improvement from the zero-shot approaches described in the paper introducing the Brainteaser dataset (Jiang et al., 2023b).

[1] https://huggingface.co/alexsherstinsky/Mistral-7B-v0.1-sharded

[2] https://www.databricks.com/blog/efficient-fine-tuning-lora-guide-llms

	Sentence Puzzle					Word Puzzle				
Model	**Avg. F1**	**F1(1)**	**F1(2)**	**F1(3)**	**F1(4)**	**Avg. F1**	**F1(1)**	**F1(2)**	**F1(3)**	**F1(4)**
GPT-4(P2)	0.783	0.817	0.833	0.810	0.334	**0.906**	**0.949**	**0.912**	0.844	0.000
GPT-4(P3)	**0.817**	**0.881**	**0.879**	**0.849**	0.571	0.865	0.914	0.889	**0.939**	0.000
FT-Mistral	0.792	0.779	0.831	0.805	**0.706**	0.844	0.853	0.862	0.808	0.000

Table 3: Average F1-scores overall and for all answer choices (1,2,3,4) for the three best performing models. It is visible that the zero-shot approaches on GPT-4, the best performing from Table 2, are biased towards the first three answers, resulting on a lower F1-score for the 4th answer (None of the above.)

Sentence Puzzles. We find that tweaking the prompt results in performance improvements. Using prompt 2 instead of prompt 1 results in a marginal increase (3%) of the overall performance and using prompt 3 results in about 6% overall improvement. In this case, the improvement is more significant in the original brainteasers with about 10%.

Word Puzzles Prompt choice seems to have a more significant impact in word puzzles. As visible in Figure 2, using prompt 2 and prompt 3 instead of prompt 1, results in respectively 26% and 30% overall accuracy increase.

4.2 Instruction-based Fine-tuning

According to Table 2 the fine-tuned model is only marginally worse that the zero-shot approaches (prompt 2 & 3) in the sentence puzzles sub-task. However, in the word puzzles sub-task, it performs 6% worse overall than the best performing zero-shot approach. In summary, the three best performing models are GPT-4 zero-shot approaches (prompt 2 & 3) and the instruction fine-tuned model.

However, due to the imbalanced distribution of correct answers between different labels, we also looked into the F1 scores of different labels. We calculated F1 scores overall and for each label and includes the results of this comparison for the three best performing models in Table 3. The table indicates that the zero-shot approaches result in lower scores in the 4th label (“None of the above”). Indeed, the F1 score for this label is improved in the fine-tuned approach. In summary, this finding highlights the need to investigate solutions and metrics beyond the simple accuracy metrics.

5 Limitations and Future Work

While we tried to explore various prompts for our zero-shot approaches, there is a possibility that further experiments might reveal more effective techniques. Future work could explore additional prompts as well as look into automating the prompt-search process. Another area of potential improvements would be the exploration of additional datasets, especially those that include similar riddles based on lateral thinking. Lastly, future work could explore additional fine-tuning techniques and discover if the accuracy can be further improved.

6 Conclusions

We present a comparison of zero-shot approaches with instruction fine-tuning within SemEval-2024 Task 9. Our experiments applied a variety of best practices for prompt engineering to explore the potential of zero-shot approaches in tasks that require lateral thinking and reasoning. We find that upon iterating over multiple prompts, zero-shots approaches using GPT-4 remain the solution that results in higher accuracy. However, instruction fine-tuned Mistral-7B provides a second best alternative in the sentence puzzle sub-task.

References

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A review on language models as knowledge bases. *arXiv preprint arXiv:2204.06031*.

Varsha Bhoir and MA Potey. 2014. Question answering system: A heuristic approach. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 165–170. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang.

2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Zijian Ding, Arvind Srinivasan, Stephen MacNeil, and Joel Chan. 2023. Fluid transformers and creative analogies: Exploring large language models' capacity for augmenting cross-domain analogical creativity. In *Proceedings of the 15th Conference on Creativity and Cognition*, pages 489–505.

Poonam Gupta and Vishal Gupta. 2012. A survey of text question answering techniques. *International Journal of Computer Applications*, 53(4).

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Nan Hu, Jiaoyan Chen, Yike Wu, Guilin Qi, Sheng Bi, Tongtong Wu, and Jeff Z Pan. 2024. Benchmarking large language models in complex question answering attribution using knowledge graphs. *arXiv preprint arXiv:2401.14640*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023b. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Nidhi Malik, Aditi Sharan, and Payal Biswas. 2013. Domain knowledge enriched framework for restricted domain question answering system. In *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–7.

Constituency Parsing. 2009. Speech and language processing. *Power Point Slides*.

Marco Antonio Calijorne Soares and Fernando Silva Parreiras. 2020. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University-Computer and Information Sciences*, 32(6):635–646.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pretrained language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9733–9740.

A Appendix A

A.1 Prompt 1

```
Please pick the best choice for the
brain teaser. Each brain teaser has
only one possible solution including
the choice none of above, answer
should only provide the choice:
Question: {}
Choice:
(A) {}
(B) {}
(C) {}
(D) {}
Answer:
```

A.2 Prompt 2

```
Below is an instruction that describes
a riddle, paired with four choices.
Choose the option that appropriately
answers the riddle.
### Riddle:
{}
### Options:
1 - {}
2 - {}
3 - {}
### Instruction:
In the end, print the number of the
correct answer between these tags:
<answer> </answer>:
```

A.3 Prompt 3

```
You are a great riddlemaster that is
very helpful in solving riddles.
Solve the following riddle:
{}
Consider each of the following
answers and provide reasons why
they are or are not correct.
If none is correct, print ”None”.
1) {}
2) {}
3) {}
In the end print the correct
answer between these tags:
<answer> </answer>
```