

uTeBC-NLP at SemEval-2024 Task 9: Can LLMs be Lateral Thinkers?

Pouya Sadeghi^{*1}, Amirhossein Abaskohi^{*2}, and Yadollah Yaghoobzadeh^{1,3}

¹Department of Electrical and Computer Engineering, University of Tehran

²Department of Computer Science, University of British Columbia

³Tehran Institute for Advanced Studies, Khatam University, Tehran, Iran

¹{pouya.sadeghi, y.yaghoobzadeh}@ut.ac.ir

²aabaskoh@student.ubc.ca

Abstract

Inspired by human cognition, Jiang et al. (2023c) create a benchmark for assessing LLMs’ lateral thinking—thinking outside the box. Building upon this benchmark, we investigate how different prompting methods enhance LLMs’ performance on this task to reveal their inherent power for outside-the-box thinking ability. Through participating in SemEval-2024, task 9, Sentence Puzzle sub-task, we explore prompt engineering methods: chain of thoughts (CoT) and direct prompting, enhancing with informative descriptions, and employing contextualizing prompts using a retrieval augmented generation (RAG) pipeline. Our experiments involve three LLMs including GPT-3.5, GPT-4, and Zephyr-7B- β . We generate a dataset of thinking paths between riddles and options using GPT-4, validated by humans for quality. Findings indicate that compressed informative prompts enhance performance. Dynamic in-context learning enhances model performance significantly. Furthermore, fine-tuning Zephyr on our dataset enhances performance across other commonsense datasets, underscoring the value of innovative thinking.¹

1 Introduction

Human cognition provides the foundational framework for understanding large language model (LLM) development, incorporating two critical thinking modes: vertical and lateral (Waks, 1997). Vertical thinking, synonymous with logical reasoning, relies on structured analysis and established principles. Conversely, lateral thinking, known for its creativity, challenges conventions and fosters innovative perspectives, enriching language processing capabilities. Recognizing and leveraging the synergy between vertical and lateral thinking are essential in maximizing LLMs’ cognitive potential.

^{*}Equal Contribution

¹Our codes and data are publicly available at: <https://github.com/Ipouyall/Can-LLMs-be-Lateral-Thinkers>

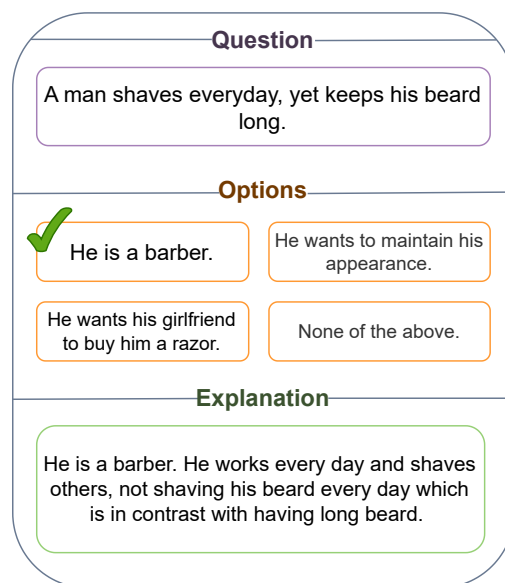


Figure 1: A sample from the sentence puzzle sub-task with an explanation of how this puzzle deprecates default commonsense.

Integrating both strategies facilitates adaptability and ingenuity in addressing linguistic challenges.

Despite LLMs’ success and the abundance of reasoning benchmarks (Ho et al., 2023; Abaskohi et al., 2023; Yasunaga et al., 2024), understanding their reasoning remains incomplete. Many benchmarks prioritize vertical over lateral thinking (Waks, 1997), inherent in LLMs’ pre-training data. Jiang et al. (2023c) introduces a challenging dataset, yet thorough analyses of prompting methods are lacking.

Building on previous research examining the impact of prompts on LLMs’ performance (Webson and Pavlick, 2022), our study aims to validate the genuine lateral understanding capability of LLMs. We participated in SemEval-2024, shared task 9, utilizing various prompts to assess LLMs’ lateral thinking abilities in the BrainTeaser multiple-choice QA task (Jiang et al., 2023c, 2024). The task focuses on the Sentence Puzzle (see Fig-

ure 1) and Word Puzzle sub-tasks², challenging common sense associations.

Our team, **uTeBC-NLP**, employs three different methods to evaluate LLMs’ lateral thinking ability: (I) chain of thoughts (CoT)-based strategies, (II) enhancing prompts with a detailed task description and prompt compression, and (III) in-context learning ability, using retrieval-augmented generation (RAG) to select dynamic samples. We conducted these experiments on three LLMs: GPT-3.5³, GPT-4⁴, and Zephyr-7B- β (Tunstall et al., 2023), a fine-tuned version of Mistral-7B (Jiang et al., 2023a) trained on a combination of publicly available, synthetic datasets using direct preference optimization (DPO) (Rafailov et al., 2024).

Our contributions include: (I) exploring the impact of incorporating task information on lateral thinking, (II) developing a thesis-based approach wherein we delineate a path between each question-option pair separately, utilizing this thesis as contextual information in subsequent runs—termed as *external-CoT*, (III) leveraging RAG for generating few-shot examples to assess the efficacy of dynamic few-shot inference (IV) employing the generated thesis context to fine-tune Zephyr-7B- β and evaluating its impact on the model’s comprehension of commonsense datasets.

In summary, our findings reveal that not all LLMs possess lateral thinking capabilities, with the ability more prominent in models with a greater number of parameters and exposure to extensive data. Proper prompting and introducing unconventional patterns would enhance this capability, by moving beyond conventional linear thinking. Models tend to prefer brief and informative prompts over lengthier alternatives. Notably, we excelled in the Sentence Puzzle sub-task, achieving a remarkable score of 0.975 in solving sentence puzzles, surpassing the baseline of 0.608, and securing the second-highest score.

2 Background

Chain of Thoughts Prompting. In-context zero-shot and few-shot learning play crucial roles in the success of LLMs. To enhance LLM performance across various tasks, including reasoning tasks, as proposed by Wei et al. (2022), we employ the CoT methodology. Replicating Wei et al. (2022)’s setup

with sample shots and ensuring their quality can be challenging, so, to ensure fairness and rely on LLMs’ knowledge, we adopt Kojima et al. (2022)’s zero-shot-CoT approach and referred to as Simple-Internal-CoT. In our simple-Internal-CoT experiments, we allow LLMs to autonomously handle the problem, thereby separating their performance from the quality of the provided samples. We introduced other variations for CoT, (I) Specific-Internal-CoT, in which we explain thinking steps to the LLM, and (II) External-CoT, in which we externally help the LLM to follow the steps by asking it to do only one step in each inference, and preserve results use in next steps’ prompts.

Enhanced Prompting Strategies. Prompt strategies, particularly CoT, prove effective in enhancing LLMs’ performance across tasks. However, our experiments demonstrate that CoT may not consistently yield better results. Another approach is boosting the model’s performance, using informative prompts, inspired from Fernando et al. (2023). We developed a detailed prompt to familiarize LLMs with diverse riddle approaches in a simple and informative manner, aiming to prevent hasty answers and help them to know how should be faced with riddles. Acknowledging the lengthiness of our detailed prompt and to prevent this factor from limiting LLMs’ performance, we created a highly compressed version following Jiang et al. (2023b), retaining essential details to improve LLMs’ performance while minimizing prompt length.

In-context Learning. In-context learning proved to be one of the most powerful methods to enhance LLMs’ performance (Brown et al., 2020). Few-shot prompting with static samples is examined by Jiang et al. (2023c) and shown not to work well. We Employed a RAG pipeline to dynamically select samples from the training split of the dataset. We focused on a three-shot manner and examined different ways of using our RAG pipeline to achieve the best performance. We also explore whether we need to explicitly mention the relation between the question and its answer or whether it can be inferred that effectively.

3 Methodology

This section begins with an overview of the dataset, followed by a detailed exposition of our methodology. We will start by elucidating the task information context. Subsequently, we will explore

²We just participated in the Sentence Puzzle sub-task.

³We used the gpt-3.5-turbo-0125 version.

⁴We used the gpt-4-0613 version.

different variations of CoT before concluding with an explanation of our RAG methods.

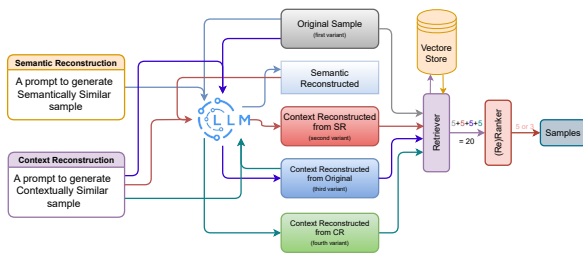


Figure 2: An illustration of our rag-fusion setup. Using an LLM, we generate four variations of the original sample to identify similar ones in the dataset, then rank them to find the closest matches. See appendix D for more details and used prompts.

3.1 Dataset

BrainTeaser. The BrainTeaser dataset (Jiang et al., 2023c) is a multiple-choice question-answering task, designed to evaluate a model’s capability for lateral thinking and its ability to challenge default commonsense associations. Created to address the gap in the NLP community’s attention towards tasks requiring implicit and intricate reasoning, the dataset relies on human-like commonsense mechanisms. The authors devised a three-step approach to create the first lateral thinking benchmark, involving data collection, distractor generation, and making adversarial examples. They produced 1,100 puzzles with detailed annotations. Assessing models’ lateral reasoning consistency, they enhanced BrainTeaser questions with semantic and contextual adjustments. Experiments with top-notch language models showed a significant performance difference from humans, particularly evident across adversarial formats, which aim to avoid cheating in scores by memorizing or previously seen examples. The dataset includes 627 samples for sentence puzzles⁵ and 492 samples for word puzzles⁶. In the case of sentence puzzles utilized in our experiments, the average number of tokens in questions is 34.88, with an average of 9.11 tokens in the answers. Our experiments are focused on the Sentence Puzzle sub-task and report our results on test split (post-evaluation phase).

Additional Datasets. We generated a dataset based on BrainTeaser’s train data that contains a thinking path between a riddle and each of its options, prompting GPT-4 and revised by authors to

⁵Train: 507, Validation: 120, Test: 120

⁶Train: 396, Validation: 96, Test: 96

avoid any bias. Then fine-tuned Zephyr on this dataset. As explored by Jiang et al. (2023c), Fine-tuning models on vertical thinking and traditional commonsense datasets can’t improve performance on BrainTeaser and the model’s lateral thinking ability. We aim to evaluate whether the fine-tuned model demonstrates an improvement in general commonsense knowledge and examine how lateral thinking ability could affect the model’s performance. We utilized SWAG (Zellers et al., 2018) and CommonsenseQA (Talmor et al., 2019) for this purpose, as they don’t need lateral thinking.

SWAG is a vast and diverse dataset designed for grounded commonsense inference, comprising over 113,000 sentence-pair completion examples sourced from internet text. Each example presents a context sentence followed by a partial continuation, prompting the selection of the most plausible completion among four choices.

CommonsenseQA (CSQA) serves as a rigorous benchmark for commonsense reasoning, featuring multiple-choice questions requiring an understanding of everyday situations, world facts, and causal relationships. Questions are associated with concepts from a large commonsense knowledge graph, interconnected through various relations, challenging models to engage in complex commonsense reasoning. Our evaluation incorporates 150 random samples from each of these datasets.

3.2 Task Informative Context

One approach to evaluating whether LLMs possess lateral thinking abilities is to prompt them explicitly for such capabilities. A key strategy involves providing hints about the task, signaling to the model that it should engage in unconventional thinking. In pursuit of this, we design three variations for task description: **(I) Simple**, which doesn’t provide any special detail and serves as a base to provide evidence of how description could affect the model’s performance, **(II) Detailed**, which would provide detailed information for the task and introduce common tricks to the LLM, and **(III) Compressed**, which is generated from the detailed variation and it just point out instead of detailed explanation. See Appendix A for more details.

3.3 Thinking Strategy

Many different thinking styles are recommended to enhance LLMs’ ability to better perform on difficult tasks. CoT prompting has emerged as a potent

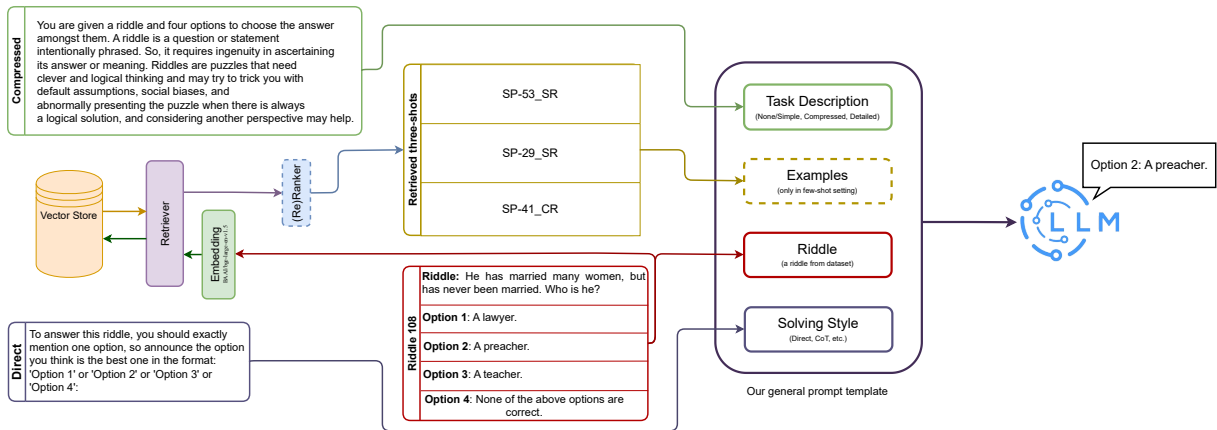


Figure 3: An overview of our approaches in solving the BrainTeaser riddles. In this setup, we have a **direct** prompt that asks the model to find the appropriate answer. To provide more information to the model, we can offer some task explanation, with the **compressed** version depicted in this figure. Finally, we utilize our RAG setup to provide the model with in-context examples. In some experiments, we also include the **theses** for each question-option pair in the prompt, serving as an unbiased link between the question and the option.

strategy for enhancing the LLMs’ performance, particularly in tasks requiring complex reasoning such as arithmetic and commonsense reasoning (Zhang et al., 2022; Diao et al., 2023; Zou et al., 2023) and known as a popular choice for these complex tasks. However, the question of how CoT should be implemented remains an open challenge. Moreover, we should be aware that CoT won’t achieve the best results in all cases, and use it or not, depends on the task and model.

We consider CoT prompting as two main approaches: (I) **Internal CoT** and (II) **External CoT**. Internal CoT involves guiding the model through step-by-step thinking or incrementally posing questions to facilitate analytical consideration of each option. Our exploration of internal CoT encompasses two types: (I) Simple, and (II) Specified. In Simple Internal CoT, the model is prompted to think step-by-step without explicit specification of each intermediate step. Specified Internal CoT provides the model with explicitly outlined steps to follow in reaching its answer. Conversely, in External CoT, similar to specified-internal-CoT, we defined steps that the model should pass to reach the final answer, but instead of letting the model control the process, we prompt it to do one step in each inference and use the model’s response to generate next prompts till we reach to the final answer. Our suggested intermediate reasoning steps, *"find a path between the question and each answer option and then select the most logical one,"* are independent for each question-option pair, and referred to as *"thesis"*. Then we would use them as context

for each option of the riddle and prompt model to solve the riddle regarding provided contexts. In Section 4.1 and Figure 4a, various CoT methods along with direct prompting, are compared.

3.4 In-context Learning

In this method, we let the model learn the task, using sample(s), known as few-shot prompting. In our few-shot experiments, we individually utilized three samples per question (Figure 3). We observed that employing static samples, as traditionally done in few-shot prompts (Brown et al., 2020), did not yield a significant performance boost, supporting few-shot results examined by Jiang et al. (2023c). To overcome this limitation, we developed a RAG pipeline to select shots dynamically based on each question. Our experiments involved three RAG methods: Ordinary RAG, Ranked RAG (RAG+ReRanker), and RAG-Fusion. Our experiments are all the same for these approaches and samples are selected from train split. Our shot instances are available with three entities: question, ground-truth answer, and explanation. Our explanations are sampled using GPT4, and they are a logical thinking path from question to answer. We also generated another variant for explanation named Summarized, which compressed long explanations. See Appendix D for more detail.

Ordinary RAG. Within the ordinary RAG approach, we employed the established RAG framework (Lewis et al., 2020) to generate contextualized representations for the question. The RAG model retrieved relevant passages from a knowl-

edge source, supplying contextual information crucial to the model’s decision-making process.

Ranked RAG. The Ranked RAG approach (Lewis et al., 2020) entailed enhancing the RAG framework by incorporating a reranking mechanism. In this variant, retrieved passages would be reranked based on their relevance to the given question, prioritizing those deemed most relevant. This integration is aimed to enhance the quality and relevance of contextual information provided to the model and GPT-4’s contexts had the most positive impact⁷ on this approach.

RAG Fusion. The RAG-Fusion method (Rackauckas, 2024) seamlessly integrates elements from both ordinary RAG and Ranked RAG. In this methodology, we generate three distinct variants derived from the original riddle, which are subsequently input into the RAG pipeline for sample retrieval. After this step, a ranker⁸ is employed to prioritize these samples (Figure 2). This multi-step process is meticulously designed to capture diverse contextual nuances and semantic variations, thereby significantly enhancing the overall effectiveness of the RAG-Fusion method. The most important weakness of this approach is that it is time-consuming as we need many LLM inferences that would take a long time.

Benchmarking. We designed a benchmark, in which we used samples from the train split, and retrieved 5 unique samples at the end, to observe each variant’s performance on different setups. for each retrieved sample from the same group, including the sample itself, the method would give one point for that sample⁹ as shown on 2 ordinary RAG can satisfy our needs in a more sample-efficient manner.

4 Experiments and Results

In this section, we report and discuss our results. Table 1 shows our submission scores during the competition. We first report and discuss our results on different methods (See Appendix C for full results). Next, we will discuss the effect of fine-tuning on our lateral thinking dataset.

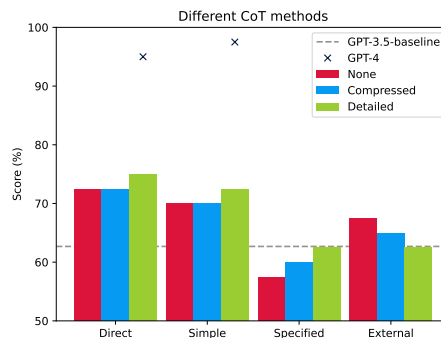
⁷Help this method to provide more helpful samples for our purpose.

⁸The same reranker used as Ranked RAG.

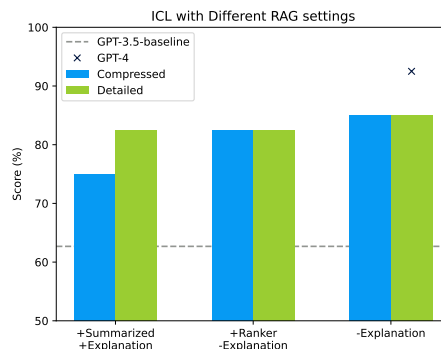
⁹Max points for one sample: 3, as we have three variations in each group.

Metric	Score (%)
Ori	97.5
Sem	87.5
Con	82.5
Ori&Sem	85.0
OriSemCon	75.0
Overall	89.2

Table 1: Our Final Submission for Sentence Puzzle sub-task. This submission was made by GPT-4 in Simple-Internal-CoT and detailed task description setting.



(a) Effect of different thinking (solving) styles on model’s performance. Bars belong to GPT-3.5.



(b) Our proposed RAG-pipelines for dynamic in-context learning(Direct Prompting) and its effect on the model’s performance. Bars belong to GPT-3.5.

Figure 4: Different prompting approaches and how they affect the model’s performance. GPT-3.5-baseline reported by Jiang et al. (2023c).

Used Samples	RAG Type	Hit Rate
20	Ordinary	0.65
	Fusion	0.767
	Ranker	0.65
507	Ordinary	0.753
	Ranker	0.73

Table 2: Results of RAG’s variants on the train split.

4.1 Prompting Methods

In this section, we detail our exploration of various prompting methods, as outlined in Section 3. Figure 4a presents a comparative analysis, revealing that, among different CoT methodologies, simple internal CoT exhibits superior performance. However, it scores lower than direct prompting, without any CoT variation employed. Notably, external CoT outperforms specifying steps in one prompt (specified internal CoT) but falls short compared to simple internal CoT and direct prompting. This is attributed to the impact of prompt length, where longer prompts with similar information weaken performance.

Prompt Length In task descriptions, providing hints consistently aided the model, with the condensed and detailed versions excelling in different scenarios. Our hypothesis, supported by the results, posits that both prompt and cognitive pathway length significantly influence performance. Extensive factors lead the model to favor concise yet informative prompts, as evidenced by the superior performance with the compressed descriptions.

In-context Learning As explained in Section 3, we focused on repeating experiments with Ordinary RAG and RAG+(Re)Ranker in three-shot format, leveraging three entities: question, ground-truth answer, and explanation.

Our explanations, sampled using GPT-4, represent a logical thinking path from question to answer. Additionally, we introduced a summarized variant, generated through Cohere’s summarize API^{10 11} for explanations exceeding 250 words. As depicted in Figure 4b, using (Re)Ranker does not significantly enhance performance. Increasing task description details improves performance. Interestingly, we observed that excluding explanation and letting the model infer the thinking path between riddles and their answer will boost LLMs’ performance, proving LLMs’ ability to extract relations and thinking paths independently.

Examining three semantically similar questions in a three-shot format, the LLMs’ performance converges to a certain score, independent of the task description. The optimal performance is achieved when excluding explanations and using a simple RAG pipeline without using (re)ranker. See Appendix D for more details on settings.

¹⁰<https://cohere.com/summarize>

¹¹Settings: length="short", extractiveness="high"

4.2 Lateral Thinking Tuning

Fine-tuning is a core strategy for refining model performance in specific tasks. However, as noted in Jiang et al. (2023c), fine-tuning on other commonsense datasets may not guarantee performance improvements; potentially, it may even lead to a decline in the overall model’s performance. This experiment focuses on fine-tuning the model using the dataset generated by GPT-4 and revised by authors, where the model discerns paths between each riddle and its options. The goal is to assess the impact of lateral thinking on model performance by evaluating it on other commonsense datasets. We tried to keep the dataset unbiased, enabling LLM to learn lateral thinking ability.

Model	Dataset	Tuned	Accuracy
Zephyr-7B- β	SWAG	No	38
		Yes	46
	CSQA	No	31.33
		Yes	36

Table 3: Fine-tuning experiments, observing model’s performance improvement in commonsense.

Table 3 showcases that fine-tuning with a lateral thinking approach significantly enhances the model’s performance on other commonsense datasets. This result challenges the conventional belief that linear thinking might constrain the model in scenarios requiring unconventional solutions. Adopting an out-of-the-box thinking approach proves beneficial, emphasizing the importance of lateral thinking across diverse contexts.

5 Conclusion

In conclusion, our study emphasizes the crucial role of prompting methods in augmenting the lateral thinking capabilities of LLMs. Through diverse CoT-based strategies, prompt refinements, and RAG techniques for in-context learning, we showcase the efficacy of well-structured prompts and thinking styles in elevating LLM performance. Additionally, fine-tuning models on a lateral thinking dataset proves advantageous, leading to improved performance on various commonsense tasks. This underscores the significance of integrating out-of-the-box thinking in model training, opening promising avenues for future research to enhance LLMs’ reasoning abilities.

References

- Amirhossein Abaskohi, Sascha Rothe, and Yadollah Yaghoobzadeh. 2023. [LM-CPPF: Paraphrasing-guided data augmentation for contrastive prompt-based few-shot fine-tuning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 670–681, Toronto, Canada. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. [Active prompting with chain-of-thought for large language models](#). *arXiv preprint arXiv:2302.12246*.
- Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. [Promptbreeder: Self-referential self-improvement via prompt evolution](#). *arXiv preprint arXiv:2309.16797*.
- Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. [Large language models are reasoning teachers](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.
- 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*.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. [LLMLingua: Compressing prompts for accelerated inference of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.
- 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. 2023c. [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.
- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Zackary Rackauckas. 2024. [Rag-fusion: a new take on retrieval-augmented generation](#). *arXiv preprint arXiv:2402.03367*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. [Direct preference optimization: Your language model is secretly a reward model](#). *Advances in Neural Information Processing Systems*, 36.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. [CommonsenseQA: A question answering challenge targeting commonsense knowledge](#). 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. [Zephyr: Direct distillation of lm alignment](#). *arXiv preprint arXiv:2310.16944*.
- Shlomo Waks. 1997. [Lateral thinking and technology education](#). *Journal of Science Education and Technology*, 6:245–255.
- Albert Webson and Ellie Pavlick. 2022. [Do prompt-based models really understand the meaning of their prompts?](#) In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In

Advances in Neural Information Processing Systems, volume 35, pages 24824–24837. Curran Associates, Inc.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. [C-pack: Packaged resources to advance general chinese embedding](#).

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H. Chi, and Denny Zhou. 2024. [Large language models as analogical reasoners](#). In *The Twelfth International Conference on Learning Representations*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. [SWAG: A large-scale adversarial dataset for grounded commonsense inference](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Anni Zou, Zhuosheng Zhang, Hai Zhao, and Xian-gru Tang. 2023. Meta-cot: Generalizable chain-of-thought prompting in mixed-task scenarios with large language models. *arXiv preprint arXiv:2310.06692*.

A Task description Prompts

During our analysis, we explored various combinations of prompting methodologies. Table A.1 presents our prompts for task description. The initial prompt solely defines what a riddle entails. In the compressed version, we provide general hints, such as avoiding bias. The final prompt includes eleven potential tricks that may occur in the question, along with instructions for the model to evade biases and consider superpower abilities used in the questions. These eleven hints were extracted from other sources.

B Generating Path Between Question and Answer

One of the methods that was used both in external CoT and step-by-step internal CoT was asking the model to generate a thinking path between a question-option pair. To do that, we asked the model to generate a path between question and option, without giving any judgment on the answer and considering every option can be an answer to the question. To prompt we used to was: "Your task is to generate a descriptive explanation from a question to an answer option. In the following, a question and

Prompt

A riddle is a question or statement intentionally phrased so as to require ingenuity in ascertaining its answer or meaning.

A riddle is a question or statement intentionally phrased so as to require ingenuity in ascertaining its answer or meaning. Riddles are puzzles that need clever and logical thinking, and may try to trick you with default assumptions, social biases, and abnormally presenting the puzzle when there are always a logical solution, and considering another perspective may help.

A riddle is a question or statement intentionally phrased so as to require ingenuity in ascertaining its answer or meaning. Different ideas can be used in riddles to trick you: 1. Riddles often employ misdirection, leading you away from the actual solution. 2. They include elements with double meanings, requiring a keen eye for words with dual interpretations. 3. Metaphorical wordplay adds another layer, urging you to decipher figurative language. 4. Look out for exaggeration, as riddles may present overly dramatic details to divert your attention. 5. Common phrases and sayings may hide within the puzzle, demanding familiarity. 6. Associations and irony play a crucial role, introducing unexpected connections. 7. Numerical puzzles can also be part of the mystery, requiring you to decode their significance. 8. Elemental imagery, drawn from nature, might hold key descriptors. 9. Rhyming and sound clues can add a poetic dimension. 10. Avoid sexism and sex cliché, for example, gender bias for jobs, based on their positions or their outcome. 11. Riddle may try to present something impossible or in contradiction with the reality. Just consider alternative perspectives.

Table A.1: Our task description prompts. The first prompt lacks task details, the second is compressed, and the last is detailed, covering all potential tricks.

an option as the answer to the question are provided. The provided option might or not be a correct answer. Write a descriptive explanation in at most one paragraph and 200 words to show the thinking path from the question to the option.." To avoid hallucination, we tried to limit the model's description to 200 by asking the model to do so. Although it does not work all the time,

it limits the model's generated words. This limitation would be later beneficial as long input prompts could decrease the model's performance.

C Complete experiments and results

In this section, we will show our complete results for the sentence puzzle sub-task. Our scores on different experiments are shown in Table C.1 and each experiment's descriptions are available in Sections 3 and 4.

D RAG extensive experiments

In this section, we discuss our rag experiments in more detail. First, we mention the common setting between different methods, then we will mention each method and explain its specific experimental setup in more detail.

Commonly, we used the Chroma¹² as our vector store. We employed "bge-large-en-v1.5" (Xiao et al., 2023) as our embedding. Our final samples are chosen as the first three unique¹³ samples retrieved from our vector store.

Our initial Explanations are selected from the same dataset used for lateral thinking tuning, using ground-truth answers instead of all options. Also for our summarized variant, for explanations longer than 250 words, we used summarizer (in this case, Cohere's summarize API) to summarize explanations.

Overall, our RAG methods can extract different variations in a group, as seen in Table 2 and Origan and SR variations seem to be closely related, but in some cases, it face problems to related CR variation into two other variations.

Ordinary RAG. We have used it as its ordinary usage. Despite having a close performance to RAG+Ranker, we decided to use this method, reducing the ranker's effect on our performance.

RAG+Ranker In this method, we first use a normal retriever as Ordinary RAG to retrieve 25 samples from our vector store. Then we fed our query and retrieved 25 samples to reranke(ranker)¹⁴ and kept the first 3 samples with the highest scores.

RAG Fusion In this method. We designed two prompts to generate semantically or contextually

related (see Table D.1). Using those two prompts, by prompting Zephyr-7B- β , we generate three new variations from the original, and counting the original sample, we feed each variation to the retriever to retrieve 5 samples, which provides 20 samples. Then we would run deduplication to eliminate duplicated samples, which may caused by employing multiple retrieval phases, and rank remained samples using a ranker(re-ranker) and keep the first five samples with the highest score (see Figure D.1). The we just continue with the first three samples as our shots.

¹²<https://github.com/chroma-core/chroma>

¹³The "unique" term comes meaningful with rag-fusion, as it may retrieve the same samples in each retrieval phase

¹⁴We used Cohere's reranker: <https://txt.cohere.com/rerank/>

Model	Thinking Method	In-Context Learning	Task Description	Result	
GPT 3.5	Direct	-	None	72.5	
			Compressed	72.5	
			Detailed	75	
	Simple-Internal-CoT	-	None	70	
			Compressed	70	
			Detailed	72.5	
	Specified-Internal-CoT	-	None	57.5	
			Compressed	60	
			Detailed	62.5	
	External-CoT	-	None	67.5	
			Compressed	65	
			Detailed	62.5	
	Simple-Internal-CoT	ES	Compressed	72.5	
	Direct	ES	Compressed	75	
	Direct	ES	Detailed	82.5	
	Simple-Internal-CoT	ER	Compressed	72.5	
	Direct	R	Compressed	82.5	
	Direct	R	Detailed	82.5	
Direct	ord	None	85		
Direct	ord	Compressed	85		
Direct	ord	Detailed	85		
Simple-Internal-CoT	ord	Compressed	77.5		
Specified-Internal-CoT	ord	Compressed	67.5		
GPT 4	Direct	-	Detailed	95	
	<u>Simple-Internal-CoT</u>	-	<u>Detailed</u>	<u>97.5</u>	
	Direct	ord	Compressed	92.5	
Zephyr-7B-β	Direct	-	None	27.5	
			Detailed	32.5	
	Simple-Internal-CoT	-	Compressed	37.5	
			Detailed	15	
			ER	Compressed	40
			ES	Compressed	42.5
			ESR	Compressed	35
			ord	Compressed	25
			R	Compressed	22.5

Table C.1: Our complete submission result for the post-evaluation phase on test split. In-context learning means using three shots dynamically selected by our RAG’s pipeline, in which: **E)** use Explanation, **S)** use Summarizer, **R)**use Ranker, and **ord)** using ordinary rag without explanation and ranker. Our final submission is underlined.

	Prompt
Semantically Related	<p>Semantic reconstruction involves rephrasing a given text while preserving its original meaning. In this context, you are presented with a riddle. The task is to rephrase the riddle without altering the correct answer. Perform a semantic reconstruction of the following riddle.</p> <p>ORG Riddle: "Four men were in a boat on the lake. The boat turns over, and all four men sink to the bottom of the lake, yet not a single man got wet! Why?"</p> <p>SR Riddle: "A boat on the lake included four men. All four men on the boat sink to the bottom of the lake when it flips over. However, not a single man gets wet! Why?"</p> <p>ORG Riddle: "A plane crashed, and every single person on board this flight was killed, yet there were survivors. Explain how?"</p> <p>SR Riddle: "Despite the fact that the entire flight was lost in a plane crash and each single person is killed, there were survivors. Describe how?"</p> <p>ORG Riddle: "{riddle}"</p> <p>SR Riddle:</p>
Contextually Related	<p>Context reconstruction involves maintaining the original reasoning path while changing both the question and the answer to describe a new situational context. In this context, you are presented with a riddle. The task is to reconstruct the context of the riddle while keeping the original reasoning intact. Ensure that the reconstructed context maintains the same reasoning path as the original riddle and also it is reasonable. You should change the context and try to avoid it by just replacing some entities and trying to convey the question to another scenario. Perform a context reconstruction of the following riddle.</p> <p>ORG Riddle: "A woman shoots her husband. Then she holds him underwater for over 5 minutes. Finally, she hangs him. But 5 minutes later, they both go out and enjoy a wonderful dinner together. How can this be?"</p> <p>CR Riddle: "A woman shoots publicly at people at a National Park. The park is full of people, but no one gets killed. How is that possible?"</p> <p>ORG Riddle: "There are 3 apples available for 2 fathers and 2 boys to consume. They each receive a single apple. How is it a mathematical possibility?"</p> <p>CR Riddle: "Two mothers and two daughters were asking for new state IDs, but the agent only gave out three forms and instructed them on how to fill them out. Why?"</p> <p>ORG Riddle: "{riddle}"</p> <p>CR Riddle:</p>

Table D.1: Prompts that are used to generate SR and CR-related samples for the RAG-Fusion method.

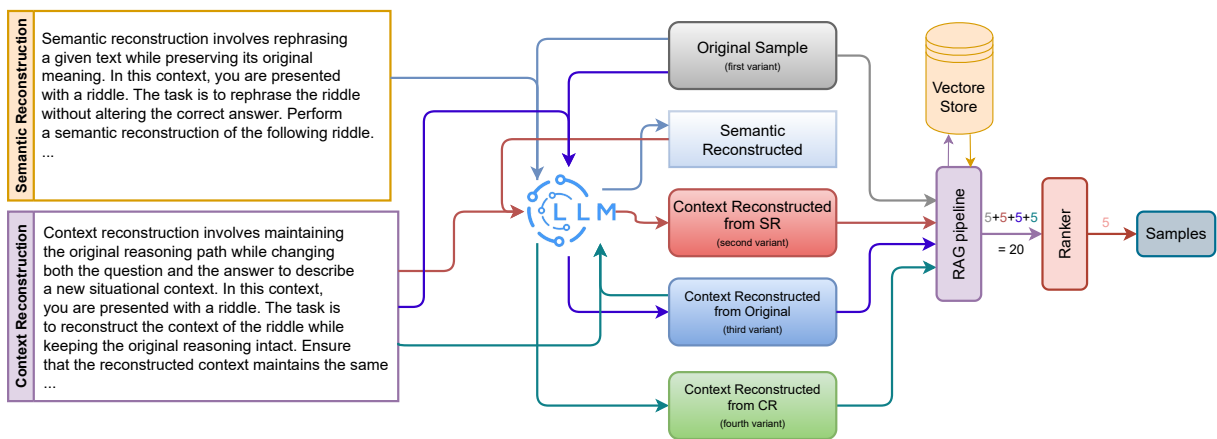


Figure D.1: RAG Fusion. The four used variants include: (I) The original riddle, (II) Context reconstruction obtained from semantically reconstructed samples originating from the original riddle, (III) Context reconstruction derived from the original riddle, (IV) Context reconstructed from step 3, then we retrieve similar samples for each variant. In the end, we feed retrieved documents to a ranker to filter them based on similarity and usefulness.