FtG-CoT at SemEval-2024 Task 9: Solving Sentence Puzzles Using Fine-Tuned Language Models and Zero-Shot CoT Prompting

Micah Zhang **Shafiuddin Rehan Ahmed** **James H. Martin**
University of Colorado, Boulder, CO, USA
micah.zhang@colorado.edu

Abstract

Recent large language models (LLMs) can solve puzzles that require creativity and lateral thinking. To advance this front of research, we tackle SemEval-2024 Task 9: BRAINTEASER: A Novel Task Defying Common Sense. We approach this task by introducing a technique that we call Fine-tuned Generated Chain-of-Thought (FtG-CoT). It is a novel few-shot prompting method that combines a fine-tuned BERT classifier encoder with zero-shot chain-of-thought generation and a fine-tuned LLM. The fine-tuned BERT classifier provides a context-rich encoding of each example question and choice list. Zero-shot chain-of-thought generation leverages the benefits of chain-of-thought prompting without requiring manual creation of the reasoning chains. We fine-tune the LLM on the generated chains-of-thought and include a set of generated reasoning chains in the final few-shot LLM prompt to maximize the relevance and correctness of the final generated response. In this paper, we show that FtG-CoT outperforms the zero-shot prompting baseline presented in the task paper and is highly effective at solving challenging sentence puzzles achieving a perfect score on the practice set and a 0.9 score on the evaluation set.

1 Introduction

The BRAINTEASER SemEval-2024 Task (Jiang et al., 2023, 2024) explores the ability of large language models (LLMs) to perform lateral thinking or "thinking outside the box", a topic that is currently under-explored by the natural language processing (NLP) community. Unlike vertical thinking tasks that rely only on "common sense", solving this task requires a creative thinking process. The goal is to force LLMs to challenge their preconceptions and consider new perspectives. The task organizers propose a way to assess the ability of LLMs to think outside the box by creating a multiple-choice question-answering task designed to defy default commons-sense associations. For this task, the task organizers created a sentence puzzle dataset that contains sentence-type brain teasers centered on sentence snippets, and a word puzzle dataset that contains word-type brain teasers centered on the letter composition of the target question. Both datasets are written in English.

Our approach, Fine-tuned Generated Chain-of-Thought (FtG-CoT), as depicted in Figure 1, is a novel few-shot prompting method that combines a fine-tuned BERT classifier encoder with zero-shot chain-of-thought (CoT) generation and a fine-tuned GPT-3.5 LLM. The BERT (Devlin et al., 2019) classifier is fine-tuned by treating the training set as a multi-class classification task. Each training set question and choice list is treated as the classification example. The index corresponding to the correct answer choice is the class label. Reasoning chains for each example in the training set are generated using a variation of the classic zero-shot chain-of-thought prompt.

For each example, in addition to the question and choice list, the LLM is provided with the correct answer and asked to generate an explanation of why that is the correct answer. The generated chains of thought are then used to fine-tune the LLM. For each example in the fine-tuning dataset, the prompt contains the question and choice list, and the response contains the generated chain-of-thought. To get the final generated response, the fine-tuned LLM is queried with a few-shot prompt that contains a set of generated reasoning chains as example demonstrations.

The set of training demonstrations provided in the few-shot prompt is chosen based on their cosine similarity to the test question. Only the top 20 most similar training demonstrations are provided, ranked in order of increasing similarity. We chose to use OpenAI's GPT-3.5 Turbo model as the pre-trained LLM due to the ability to easily query and

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1245–1251

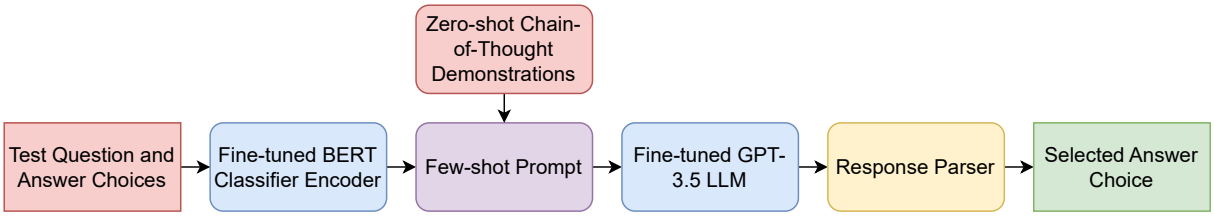

Figure 1: System diagram for FtG-CoT. To start, the test question and choice list are fed to a fine-tuned BERT classifier. The BERT classifier outputs an encoding that is used to identify the top 20 training questions that are most similar to the test question. The corresponding training demonstrations are combined in order of increasing similarity with the test question and choice list to create a few-shot prompt. The few-shot prompt is sent to a fine-tuned LLM, which outputs a generated response. A text parser matches the generated response with the most similar answer choice from the choice list.

fine-tune the model via the OpenAI API.

We provide a robust system that combines fine-tuned BERT and GPT-3.5 together with zero-shot CoT generation and ranked few-shot prompting. We found that this approach significantly improves the ability of the LLM to solve brain teaser sentence puzzles compared to the baseline zero-shot prompting approach, achieving a perfect score on the practice set and a 0.9 score on the evaluation set. Our method ranked 18th in the competition. We provide our code here: https://github.com/Micah-Zhang/SemEval-2024

2 Background

The BRAINTEASER task (Jiang et al., 2023, 2024) consists of two sub-tasks: Sentence Puzzles and Word Puzzles. Both datasets are written in English and each question is structured in a multiple-choice question format. Sentence Puzzles tend to be in a longer narrative story format. The correct answer challenges the common-sense interpretation of the question and the common-sense answer to the question. For Word Puzzles, the correct answer challenges the default meaning of a particular word and focuses on the letter composition of the question. Both types of questions provide 4 different answer choices, including 1 correct answer and 3 distractors. Figure 2 provides an example of a Sentence Puzzle and an example of a Word Puzzle for comparison.

We chose to participate in the Sentence Puzzle sub-task. For this sub-task, a training dataset consisting of 507 sentence puzzles was provided by the task organizers. The training set contained 169 original sentence puzzle examples obtained from public websites via web crawlers. Each example consisted of a question, a list of answer choices, the correct answer, and the distractors in the choice list. For each original question, a corresponding semantic reconstruction question and a context reconstruction question were generated. Semantic Reconstruction rephrases the original question without changing the correct answer and provided answer choices. Context Reconstruction changes both the question and answer to fit a new situational context, while keeping the original premise intact. The two adversarial question counterparts were created by the organizers to prevent an LLM from being able to win the competition using memorization only.

In addition to the training set, the organizers provide a practice evaluation and an official evaluation dataset for the Sentence Puzzle sub-task. Both datasets contain 120 questions. To evaluate a model, a text file containing the answer choice index for each selected answer choice is submitted to CodaLab. CodaLab then automatically calculates the corresponding accuracy scores and posts the results on the leaderboard.

Our approach is inspired by Wei et al. (2023), who introduced chain-of-thought prompting and demonstrated that providing a LLM with a series of intermediate reasoning steps improves its ability to perform complex reasoning. Our approach was also inspired by Kojima et al. (2023), who introduced zero-shot chain-of-thought prompting and demonstrated that LLMs could be made to generate their own series of intermediate reasoning steps by adding the words "Let's think step by step" to the end of the prompt. BERT (Devlin et al., 2019) is a popular transformer architecture commonly used as an sentence encoder for NLP tasks. The pre-trained BERT model can be fine-tuned by adding a classifier head to the end of the model and training it on a classification dataset. LLMs based on the GPT-3.5 architecture were popularized by Brown et al. (2020), who introduced few-shot prompting

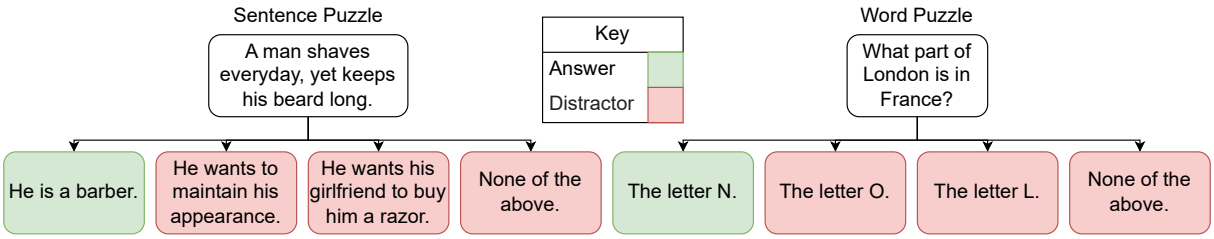

Figure 2: Diagram comparing Sentence Puzzles to Word Puzzles from the BRAINTEASER dataset. Sentence Puzzles tend to be in a longer narrative story format. The correct answer challenges the common-sense interpretation of both the question and its answer. For Word Puzzles, the correct answer challenges the default meaning of a word and focuses on the letter composition of the question.

and demonstrated that pre-trained LLMs could be made to solve new unseen tasks without needing additional training if the prompt included a set of example demonstrations.

3 System Overview

Fine-tuned Generated Chain-of-Thought Prompting (FtG-CoT) is a few-shot prompting method we developed for this competition. It consists of five steps. The first step is to fine-tune a pre-trained BERT-Small model on the training set as a multi-class classification task to use as an encoder. The second step is to use zero-shot chain-of-thought prompting (Kojima et al., 2023) to generate explanations for each demonstration in the training set. The third step is to fine-tune a large language model (LLM) on the generated chain-of-thought (Wei et al., 2023) demonstrations. The fourth step is to rank each encoded training demonstration based on its cosine similarity with the current encoded test question. The fifth step is to construct a few-shot (Brown et al., 2020) prompt by stacking the top N most similar demonstrations with the test question prompt in order of increasing similarity.

This few-shot prompt is sent to the LLM, an answer choice is extracted from the LLM's response, and the process is repeated for each question in the test set. This system is illustrated in Figure 1.

Fine-tuning BERT: We fine-tuned a pre-trained BERT model by converting it to a multi-class classifier and training it on the provided BRAINTEASER training set. This allowed us to create an encoder purpose-suited for solving sentence puzzles. During training the BERT classifier learns contextual and semantic information that is relevant to solving the sentence puzzle task. When used to encode a training or test example, this information is captured in the encoding and aids in selecting highly relevant and useful demonstrations to use for few-shot prompting.

Zero-shot prompting: Zero-shot prompting is used to generate chain-of-thought reasoning explanations for each training example. This is accomplished by prompting the LLM using a custom zero-shot prompt for each training example. The first three lines of the prompt contain the training question, answer choices, and correct answers. The fourth line is "Let's think step by step. What is the best answer? Explain in detail why this is the best answer in 5 or less sentences".

A custom fuzzy logic answer extractor parses the generated response from the LLM by isolating the sentence containing the selected answer and then using a sequence matcher to compare the generated answer against each of the answer choices in the choice list. The index of the answer choice that most closely matches the generated answer is set as the predicted label.

Fine-tuning GPT-3.5: Fine-tuning a pre-trained GPT-3.5 model on the 507 training examples and their corresponding generated reasoning chains improves the accuracy and consistency of the answers generated by the LLM. The fine-tuning dataset contains an example prompt and response for each training question. The first two lines of the prompt contain the question and answer choices. The third line is "Let's think step by step. What is the best answer?". The ground truth response is the generated response from the previous zero-shot prompting step.

Ranking Demonstrations: Cosine similarity is defined as $cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$, where A represents the vector encoding of a test question and B represents the vector encoding of a training question, both produced using the fine-tuned BERT classifier. Providing the top N training examples that have the highest cosine similarity with test questions as few-shot demonstrations ensures that the selected

demonstrations are highly relevant to the current test question.

Few-shot Prompt: The final prompt sent to the fine-tuned LLM contains all N demonstrations ranked in increasing order of similarity, the test question and answer choices, as well as the line "Let's think step by step. What is the best answer?". The received response is parsed using the same fuzzy logic sequence-matching answer extractor.

4 Experimental Setup

Training FtG-CoT requires fine-tuning BERT and fine-tuning GPT-3.5. We use all 507 examples from the sentence puzzle training set provided for the BRAINTEASER task for fine-tuning.

To fine-tune BERT, we start with the BERT-Small pre-trained model from Google (Bhargava et al., 2021) and add a multi-class classifier head consisting of two linear hidden layers separated by a ReLU activation function to create a BERT classifier. The input to the classifier head is the [CLS] token from BERT: the first token from BERT's final hidden layer. The first hidden layer in the classifier head projects the BERT output from dimensions 512 to 256. The output of the second hidden layer is passed through a log softmax function.

To train the BERT classifier on the Sentence Puzzle training set, we first concatenate the question and choice list from each example and provide them as the input string to the classifier. The index of the correct answer from the choice list is the label for the classifier. During training the BERT classifier encodes each input string via the pre-trained BERT model and passes the encoding to the classifier head, which then assigns a class label to the input. The model calculates the negative log-likelihood loss between the predicted and true label and back-propagates the loss through the entire model, including the BERT layers. The BERT classifier was trained using an AdamW optimizer for 40 epochs with a learning rate of 0.00001. The final layer of the trained BERT classifier produces encodings that have been fine-tuned to solve sentence puzzles.

To fine-tune GPT-3.5, we start with the gpt-3.5-turbo-1106 pre-trained model from OpenAI and fine-tune the model using the OpenAI fine-tuning API. We create a fine-tuning dataset according to the format required by the OpenAI API. For each of the 507 training examples the system is set to "teacher", and the user prompt is defined as a concatenation of the question and choice list. The ground truth assistant response is defined as the generated chain of thought. The fine-tuned GPT-3.5 model was trained for 3 epochs with a batch size of 1 and a learning rate multiplier of 2. All other training hyperparameters cannot be set externally by the user and are instead defined internally by OpenAI.

The model is evaluated using the test set provided for the BRAINTEASER task. The test set consist of 120 questions, each with their own choice list. For each of the 120 questions, we use FtG-CoT to create a few-shot prompt used to query the fine-tuned GPT-3.5 model via the OpenAI API. We set the temperature to 1.0. The corresponding received response is then parsed using the custom fuzzy logic answer extractor described earlier. The predicted labels for all 120 test questions are then submitted to the BRAINTEASER CodaLab website where it is automatically graded against the ground truth labels. The resulting accuracy score is displayed on the competition leaderboard.

5 Results

The task organizers created 6 different accuracy metrics for evaluation. The first 3 metrics are instance-based accuracy scores that measure the accuracy of the model in solving the original questions, the semantic reconstruction questions, and the context reconstruction questions separately from one another. The next 2 metrics are group-based accuracy scores that consider each original puzzle and its variants as a single group. For each group, the model will score 1 accuracy point only if it successfully solves all three puzzles in the group. Otherwise, it will score 0 points. The last metric is an overall accuracy that is calculated as the average of the 3 instance-based accuracy scores.

FtG-CoT performs well at the task according to the official metrics, achieving a perfect score on the practice set and a 0.9 score on the evaluation set, ranking 18th overall in the competition. In this context, "score" refers to the accuracy score for the original practice and evaluation questions, not including the semantic and context reconstruction questions.

Table 1 compares the official leader board results for FtG-CoT evaluated on sentence puzzle evaluation dataset against the zero-shot ChatGPT baseline provided by the BRAINTEASER task organizers. FtG-CoT outperformed the zero-shot baseline for

Method	Original	Semantic	Contextual	Ori + Sem	Ori + Sem + Con	Overall
Zero-shot	0.608	0.593	0.679	0.507	0.397	0.627
FtG-CoT	0.900	0.825	0.775	0.800	0.675	0.833

Table 1: Official leader board results for sentence puzzle evaluation dataset. Compares FtG-CoT performance against the zero-shot ChatGPT baseline provided by the BRAINTEASER task organizers. The six categories correspond to the three instance-based accuracy scores, the two group-based accuracy scores, and the overall accuracy score. FtG-CoT outperforms the baseline in each of these metrics.

all 6 metrics. It performed best at answering the original question and struggled the most with the combined semantic and original group-based accuracy metric.

Method	Score (Original)
1-shot	0.8
5-shot	0.925
10-shot	0.925
10-shot w/ reversed order	0.90
20-shot	0.925
10-shot w/ fine-tuning	0.975
20-shot w/ fine-tuning	1.0
30-shot w/ fine-tuning	0.975

Table 2: Experimental results comparing the performance of FtG-CoT on the sentence puzzle practice dataset with different configurations. In general, increasing the number of demonstrations, listing them in order of increasing similarity, and fine-tuning the LLM on the training demonstrations tended to result in higher accuracy scores.

Table 2 displays practice test set scores achieved by FtG-CoT using different numbers of demonstrations, ordering, and with and without fine-tuning GPT-3.5 on the training set. By default, all demonstrations were concatenated in order of increasing similarity with the test question. In general, increasing the number of few-shot training demonstrations provided in the prompt tends to improve the performance of the LLM. However, past a certain number, in our case 10 demonstrations, adding additional demonstrations does not improve the performance of the LLM and results in decreased performance.

Furthermore, reversing the order of the demonstrations to be in order of decreasing similarity such as that the least similar demonstration is located closest in proximity to the test question in the prompt resulted in a lower score. This suggests that listing few-shot demonstrations in order of increasing similarity is the better approach.

The results also illustrate the effectiveness of fine-tuning the LLM on the training set. Whereas without fine-tuning the score peaked at around 0.925 regardless of the number of demonstrations provided, fine-tuning the LLM immediately resulted in a jump in performance to 0.975. Fine-tuning the LLM combined with increasing the number of provided demonstrations to 20 resulted in the highest achieved score of 1.0. However, increasing the number of provided demonstrations past 20 did not result in additional improved performance.

6 Error Analysis

For error analysis, we performed a 80/20 randomized train/test split on the sentence puzzle training set to create our own error analysis test set. We then fine-tuned FtG-CoT on the remaining training examples and evaluated the model against this test set.

Early analysis revealed that many of the questions that the model answered incorrectly were false negatives resulting from using exact string matching for answer extraction. This motivated the creation of the fuzzy logic answer extractor. We also found that the LLM struggled with answer consistency and repeatability. Fine-tuning the model helped to lessen this issue. However, even with the fuzzy extractor and fine-tuning, the LLM was not able to achieve perfect accuracy on the error analysis test set.

Figure 3 provides both an example of a sentence puzzle that the model answered correctly as well as an example of a sentence puzzle that the model answered incorrectly. Both questions belong to the same group of sentence puzzles. The question the model answered correctly was the original sentence puzzle. The question the model answered incorrectly was the semantically reconstructed version of the sentence puzzle. The semantically reconstructed question rephrases the original question, keeps the same correct answer, and shuffles the provided answer choices.

In this case, both questions share the same premise: a group of animals are racing to grab

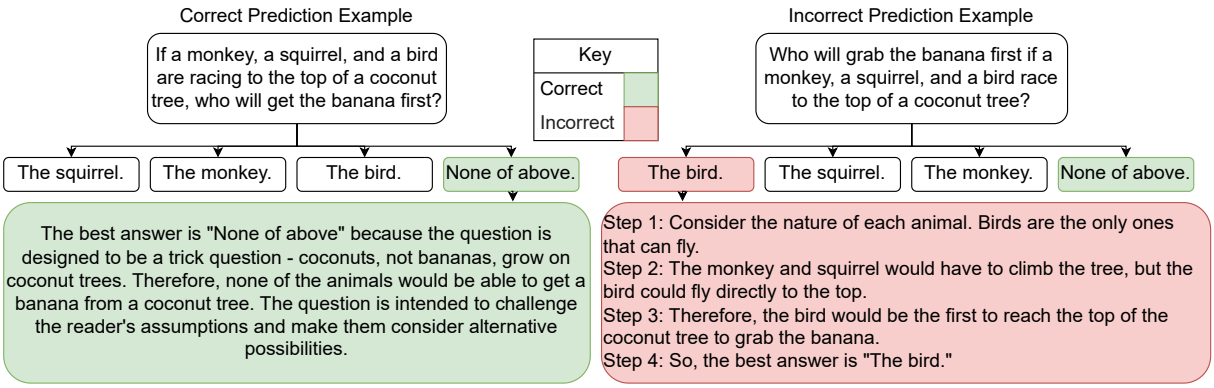

Figure 3: Example of a generated response with the correct answer for an original sentence puzzle as well as an example of a generated response with the incorrect answer for its corresponding semantically reconstructed sentence puzzle. The LLM correctly identifies the puzzle twist behind the original sentence puzzle and uses it to arrive at the correct answer, but is unable to do so for the semantically reconstructed sentence puzzle.

a fruit at the top of a tree, which animal will reach the fruit first? They also share the same puzzle twist: we are told that the animals are climbing a coconut tree yet are asked which animal will reach the banana first. Since bananas cannot grow on coconut trees, none of animals will be able to reach the banana. Therefore, the correct answer is "None of the above".

The generated response for the original sentence puzzle demonstrates that the model was able to correctly identify the puzzle twist and use it to arrive at the correct answer. However, the generated response for the semantically reconstructed question demonstrates that the model was not able to apply this same reasoning to the rephrased question. Instead, it answers the question as if it was asking about a coconut and a coconut tree and bases its answer off of the speed and movement of the animals, choosing the bird for its ability to fly.

These examples illustrate how the fine-tuned LLM model is highly sensitive to word choice and order, which helps explain why the model performs best on the original sentence puzzles and tends to struggle with the semantically and contextually reconstructed versions of the original puzzles. It is possible that the dataset OpenAI used to train GPT-3.5 contains a portion of the original sentence puzzles in the error analysis test set, allowing it to rely on memorization to improve its performance. However, since the LLM's accuracy on the reconstructed questions is only around 10% lower compared to the original questions, this suggests that the model is not relying solely on memorization to solve the sentence puzzles and may be capable of lateral thinking.

7 Conclusion

The FtG-CoT few-shot prompting method we developed for this competition combines a fine-tuned BERT classifier encoder with zero-shot chain-of-thought generation and a fine-tuned LLM. Our experiments have demonstrated that FtG-CoT is highly effective at solving sentence puzzles, achieving a perfect score on the practice set and a 0.9 score on the evaluation set, ranking 18th overall in the competition.

The key takeaways from our experiments are that FtG-CoT performs better with a larger number of demonstrations up to a certain point, few-shot demonstrations should e listed in order of increasing similarity, fine-tuning results in markedly improved performance, and that FtG-CoT significantly outperforms zero-shot prompting on the sentence puzzle evaluation set.

Regarding future work, it is possible that the performance of FtG-CoT can be further improved by using data augmentation techniques to expand the training set. For example, prompting an LLM to rephrase the existing training questions could increase the size of the training set and potentially decrease the fine-tuned model's sensitivity to word choice and order. It is also possible that the performance of FtG-CoT can be further improved by incorporating human feedback via reinforcement learning. A human could provide feedback on the quality and accuracy of the generated reasoning chains as well as manually rewrite incorrectly generated reasoning chains, improving the quality of both the fine-tuning dataset and the demonstrations.

8 Acknowledgements

We would like to thank Jie Cao and Abteen Ebrahimi for their guidance and feedback. We would also like to extend our thanks to Yifan Jiang, Filip Ilievski, and Kaixin Ma for creating the BRAINTEASER SemEval-2024 Task.

References

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

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. 2023. 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. 2023. Large language models are zero-shot reasoners.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.