

Exploring Chain of Thought Techniques

During this summer course, in the Reading Paper phase, our group was assigned the paper "Chain-of-Thought reasoning without prompting". After reading this paper, I developed an interest in the Chain of Thought (CoT) approach. Therefore, in my Individual Report, I would like to further explore and summarize the literature related to Chain of Thought.

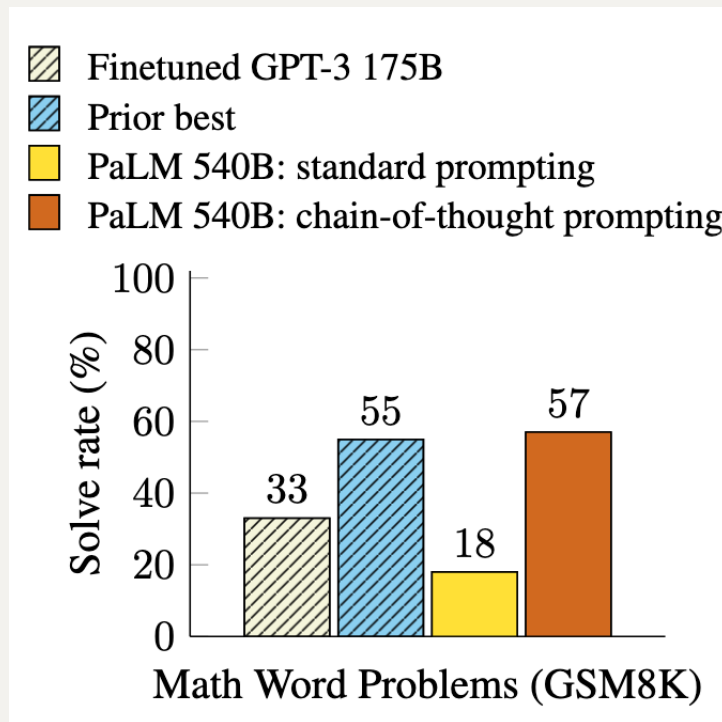
What is Chain of Thought ?

Chain of Thought is a series of short sentences that mimic the reasoning process a person might go through when answering a question. CoT Promoting is a method that provides a series of prompts to guide the model through a series of thoughts. In my view, CoT Promoting offers a paradigm for thinking when answering questions—similar to the steps we take when solving problems—except here, the process of thinking is handed over to the model to generate answers.

Compared to Zero Shot learning, CoT Promoting has several advantages:

- First, CoT provides explainability. When we receive an answer, we not only get the solution but also understand how it was derived.
- Secondly, CoT breaks down complex problems into simpler steps, enhancing the accuracy of the answers.

Experiments on three large language models show that chain-of-thought prompting improves performance on a range of arithmetic, commonsense, and symbolic reasoning tasks.



PaLM 540B uses chain-of-thought prompting to achieve new state-of-the-art performance on the GSM8K benchmark of math word problems.

How to Elicit Chain of Thought ?

Through reviewing relevant literature, I believe there are two main methods to elicit the Chain of Thought: one is through prompting, and the other is by changing the decoding approach.

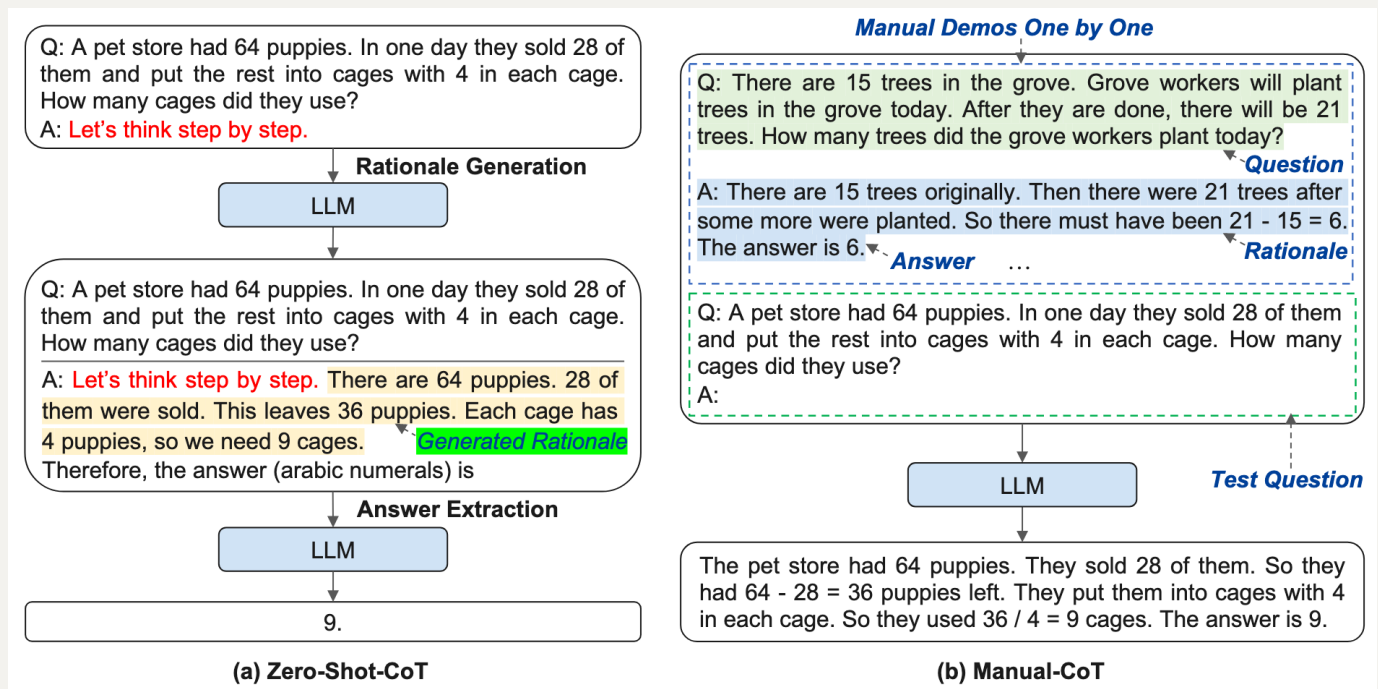
Depend on Prompting

Let's first look at how to elicit Chain of Thought through prompting. After reviewing the literature, I have identified several approaches. Here, I will mainly introduce two methods: Auto-CoT and Active Prompting.

Auto CoT

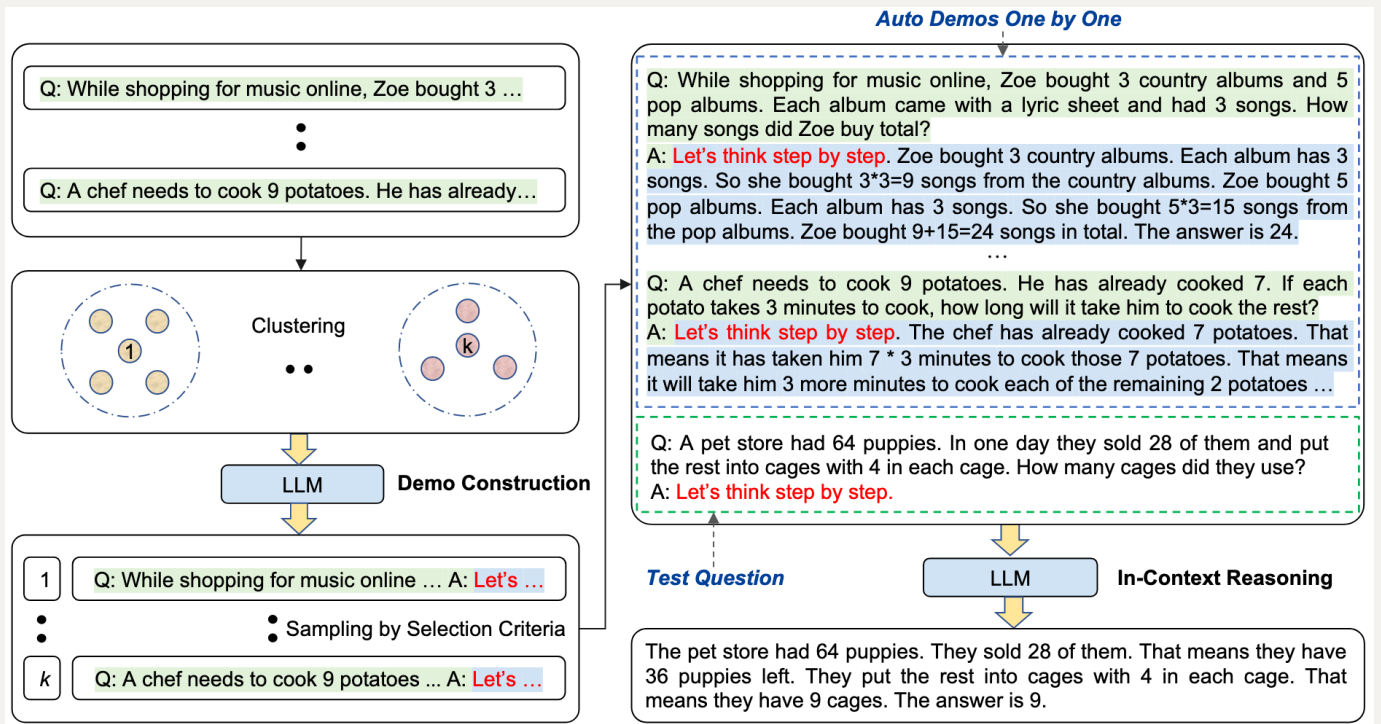
CoT prompting can be divided into two major paradigms. The first paradigm, known as Zero-Shot-CoT, involves adding a simple prompt such as "Let's think step by step" following the test question to facilitate reasoning chains in Large Language Models (LLMs). This approach does not require input-output demonstrations and is task-agnostic.

The second paradigm, Manual-CoT, relies on manually designed demonstrations. Each demonstration includes a question followed by a reasoning chain that leads to the answer. Although Manual-CoT has demonstrated superior performance, it requires significant manual effort to design task-specific demonstrations.



However, manually designing CoT prompting requires a significant amount of manpower and is not suitable for all tasks. Therefore, the authors have proposed Auto CoT, an automated method for designing CoT prompting. Briefly, Auto-CoT consists of two main steps:

1. Partition questions of a given dataset into eight clusters** — sentence-BERT is used to encode the questions, and then clusters are formed based on cosine similarity.
2. Select a representative question from each cluster and generate its reasoning chain using Zero-Shot-CoT with simple heuristics — the heuristics involve not selecting a question with more than 60 tokens or a rationale with more than five reasoning steps. These heuristics aim to improve the likelihood of the auto-generated response being correct.



In fact, the authors initially attempted to generate a series of reasoning explanations using the Zero-Shot-CoT approach. However, they found that this simple method could not effectively replace the benefits brought by manual design, as errors often occurred in the reasoning chains generated by Zero-Shot-CoT.

Further analysis revealed that the diversity of problems in the reasoning explanations is crucial for mitigating the errors caused by Zero-Shot-CoT.

This is also why the article emphasizes the approach of "Let's think not just step by step, but also one by one."

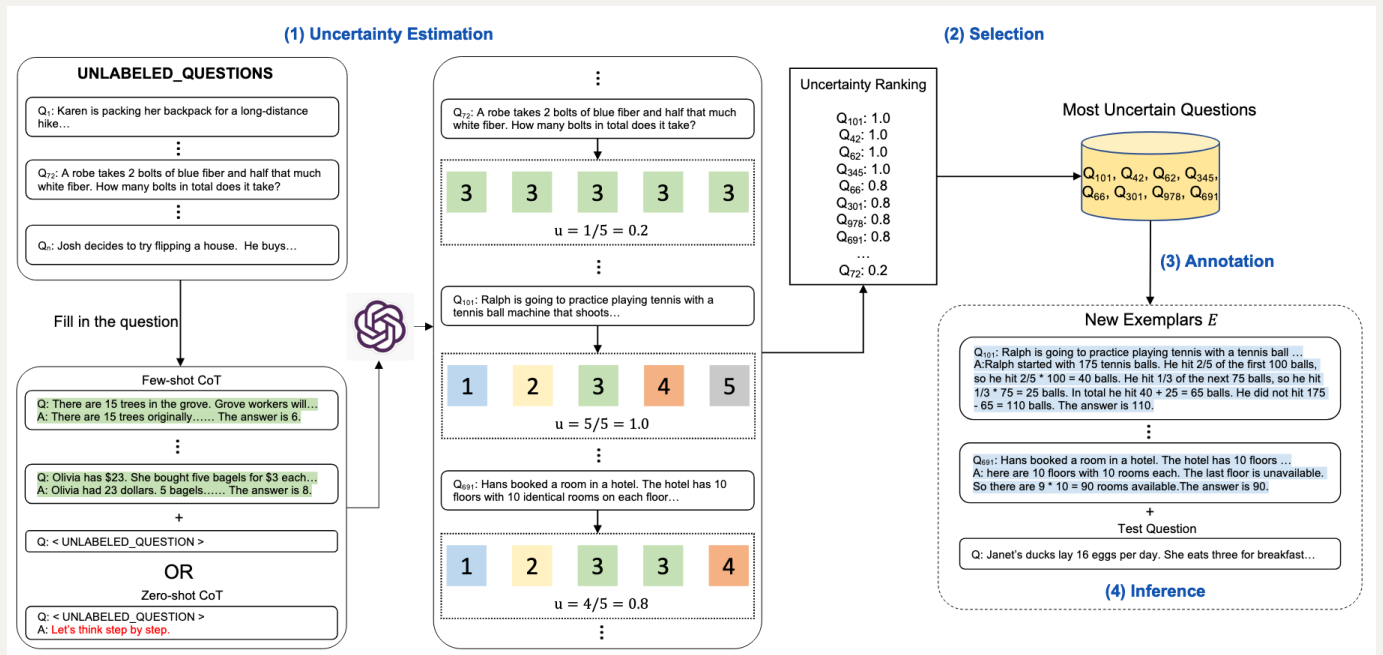
Active Prompting

In this method, the focus is on how to select prompts for Chain of Thought (CoT). Simply put, it's similar to how during an exam, we identify the questions that confuse us the most and prioritize learning them, rather than randomly choosing questions to study. This approach helps achieve better results.

The authors employed four uncertainty metrics—disagreement, entropy, variance, and confidence, with disagreement and entropy being the main metrics used in their experiments.

Subsequently, the questions identified as the most uncertain based on these metrics are selected for manual annotation.

These 'n' uncertain questions are annotated by human annotators to generate CoT rationales and then used as examples for few-shot prompting along with the test question to generate responses.



Overall, the approach of **Auto CoT** is based on diversity. It involves partitioning questions into clusters and generating reasoning chains for representative questions from each cluster. This method utilizes sentence embeddings and cosine similarity to ensure a wide coverage of different types of problems within a dataset, aiming to generate a broad spectrum of reasoning paths that enhance the model's ability to tackle diverse challenges.

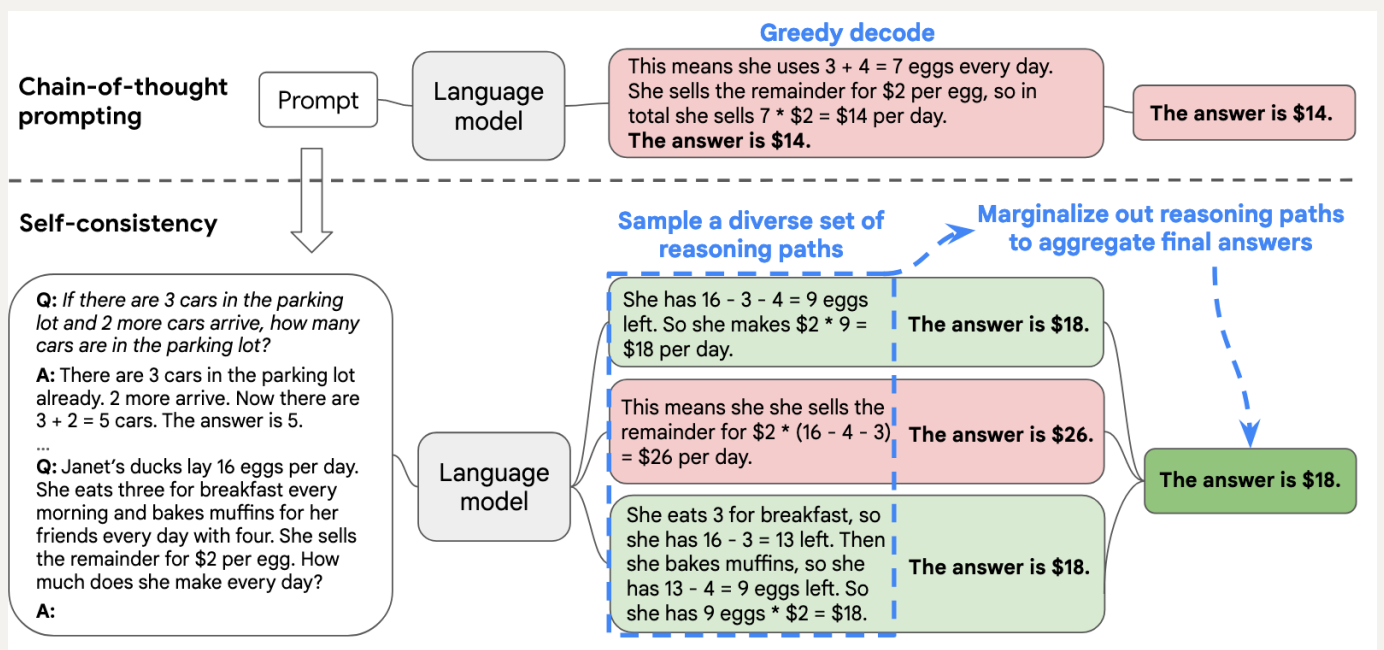
On the other hand, **Active Prompting** is based on uncertainty. It prioritizes the selection of prompts by identifying questions with high levels of uncertainty, using metrics like disagreement and entropy. These uncertain questions are then manually annotated to create detailed CoT rationales. This approach ensures that the model is trained on the most challenging aspects of a dataset, which may require nuanced reasoning or exhibit higher error rates in automated reasoning processes.

Both methods aim to improve the reasoning capabilities of language models by guiding them in how to think through problems, but they approach the task from different angles —Auto CoT through the lens of problem diversity, and Active Prompting through the lens of targeting uncertainty.

Depend on Decoding

The traditional decoding method is greedy decoding. Greedy decoding is a simple method for generating sequences where, at each step, the word with the highest probability is selected as the output, until an end symbol is generated or a predetermined output length is reached. The advantages of greedy decoding are its simplicity and speed, as it only requires selecting the highest probability word at each step. However, this method has a significant drawback: it may miss the globally optimal sequence. Since it selects only the locally optimal word at each step, greedy decoding can lead to lower quality text generation, lacking in coherence and diversity.

Self Consistency Decoding



Self-Consistency Decoding essentially aggregates the answers by selecting the most frequently occurring result as the answer. First, a set of candidate outputs is sampled from the language model's decoder, generating a group of different candidate reasoning paths. Then, the answers are aggregated by marginalizing the sampled reasoning paths and selecting the most consistent answer from the generated results.

	Method	CSQA	StrategyQA	ARC-e	ARC-c	Letter (4)	Coinflip (4)
	Previous SoTA	91.2^a	73.9 ^b	86.4 ^c	75.0 ^c	N/A	N/A
UL2-20B	CoT-prompting	51.4	53.3	61.6	42.9	0.0	50.4
	Self-consistency	55.7 (+4.3)	54.9 (+1.6)	69.8 (+8.2)	49.5 (+6.8)	0.0 (+0.0)	50.5 (+0.1)
LaMDA-137B	CoT-prompting	57.9	65.4	75.3	55.1	8.2	72.4
	Self-consistency	63.1 (+5.2)	67.8 (+2.4)	79.3 (+4.0)	59.8 (+4.7)	8.2 (+0.0)	73.5 (+1.1)
PaLM-540B	CoT-prompting	79.0	75.3	95.3	85.2	65.8	88.2
	Self-consistency	80.7 (+1.7)	81.6 (+6.3)	96.4 (+1.1)	88.7 (+3.5)	70.8 (+5.0)	91.2 (+3.0)
GPT-3 Code-davinci-001	CoT-prompting	46.6	56.7	63.1	43.1	7.8	71.4
	Self-consistency	54.9 (+8.3)	61.7 (+5.0)	72.1 (+9.0)	53.7 (+10.6)	10.0 (+2.2)	75.9 (+4.5)
GPT-3 Code-davinci-002	CoT-prompting	79.0	73.4	94.0	83.6	70.4	99.0
	Self-consistency	81.5 (+2.5)	79.8 (+6.4)	96.0 (+2.0)	87.5 (+3.9)	73.4 (+3.0)	99.5 (+0.5)

Self-consistency significantly enhances arithmetic reasoning performance across all four language models, surpassing the results achieved with chain-of-thought prompting. This approach yields substantial gains across the models and has achieved state-of-the-art results on five out of six tasks.

CoT Decoding

CoT-Decoding builds upon Self-Consistency by introducing a confidence metric. This method calculates the confidence level for each reasoning path and aggregates paths that lead to the same answer, ultimately selecting the path with the highest confidence as the final reasoning path. The concept mirrors the elimination method used in multiple-choice tests, where incorrect options are first excluded to identify the most probable answer from the remaining choices.

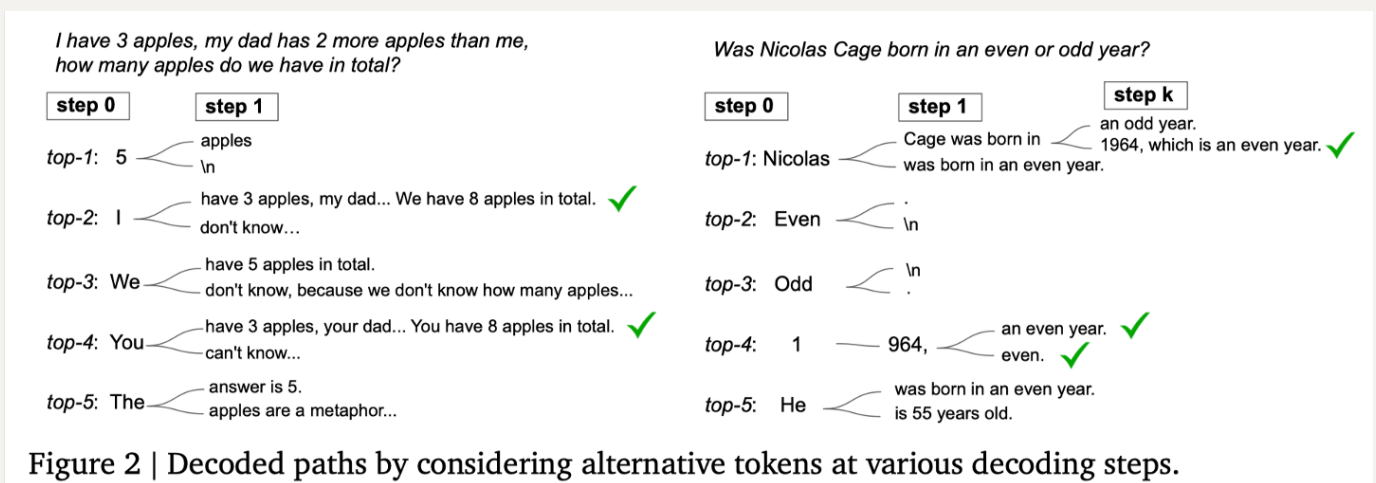


Figure 2 | Decoded paths by considering alternative tokens at various decoding steps.

This approach is particularly effective when there is a significant difference between the options. When options are closely matched, decision-making becomes challenging, and the accuracy of decisions decreases—akin to scenarios in decoding where the probability differences between tokens are minimal. In contrast, when the disparity between answer options is marked, decision-making confidence increases, similar to when there is a significant difference in token probabilities during decoding. CoT-Decoding enhances confidence assessment by accumulating the difference in probabilities between the top-2 predicted tokens.

	GSM8K Acc
Top- k sampling ($k = 10$)	4.9%
Top- p / Nucleus sampling ($p = 0.9$)	6.4%
Beam search ($b = 10$)	6.7%
Temperature sampling ($T = 0.7$)	7.5%
Greedy decoding	9.9%
Self-consistency w/o CoT prompt (10 paths)	12.9%
CoT-decoding ($k = 10$)	25.1%

In the final part of the study, the authors compared various sampling and decoding methods in their experiments. The results indicate that CoT-Decoding achieved the best performance across all tasks, demonstrating that CoT-Decoding is an effective decoding method that can enhance the performance of models.

Conclusion

This report has explored the Chain of Thought (CoT) approach, focusing on enhancing the reasoning capabilities of language models through diverse prompting and decoding strategies. **Auto CoT** and **Active Prompting** refine how models generate and improve answers by harnessing problem diversity and addressing uncertainty, respectively. Meanwhile, decoding techniques like **Self-Consistency Decoding** and **CoT Decoding** improve answer accuracy and confidence by emphasizing the most consistent and probable outcomes.

References

Diao, S., Wang, P., Lin, Y., Pan, R., Liu, X., & Zhang, T. (2024). *Active Prompting with Chain-of-Thought for Large Language Models* (No. arXiv:2302.12246). arXiv.

<https://doi.org/10.48550/arXiv.2302.12246>

Kim, S., Joo, S. J., Kim, D., Jang, J., Ye, S., Shin, J., & Seo, M. (2023). *The CoT Collection: Improving Zero-shot and Few-shot Learning of Language Models via Chain-of-Thought Fine-Tuning* (No. arXiv:2305.14045). arXiv. <https://doi.org/10.48550/arXiv.2305.14045>

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., & Zhou, D. (2023). *Self-Consistency Improves Chain of Thought Reasoning in Language Models* (No. arXiv:2203.11171). arXiv. <https://doi.org/10.48550/arXiv.2203.11171>

Wang, X., & Zhou, D. (2024). *Chain-of-Thought Reasoning Without Prompting* (No. arXiv:2402.10200). arXiv. <https://doi.org/10.48550/arXiv.2402.10200>

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2023). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models* (No. arXiv:2201.11903). arXiv. <https://doi.org/10.48550/arXiv.2201.11903>

Zhang, Z., Zhang, A., Li, M., & Smola, A. (2022). *Automatic Chain of Thought Prompting in Large Language Models* (No. arXiv:2210.03493). arXiv.

<https://doi.org/10.48550/arXiv.2210.03493>