Mitigating Misleading Chain-of-Thought Reasoning with Selective Filtering

Yexin Wu[1], **Zhuosheng Zhang**[2,*], **Hai Zhao**[3,4,*]
[1]College of Zhiyuan, Shanghai Jiao Tong University
[2]School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University
[3]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[4]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University
`{wuyexin_libro_i131, zhangzs}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn`

Abstract

Large language models have manifested remarkable capabilities by leveraging chain-of-thought (CoT) reasoning techniques to solve intricate questions through step-by-step reasoning chains. Despite its success, the efficacy of such reasoning is inherently contingent upon the quality of CoT. However, flawless CoT reasoning cannot be guaranteed due to the presence of indecomposable questions and the potential for erroneous reasoning chains, particularly in the case of small-scale language models. To tackle this challenge, we propose a novel approach called the selective filtering reasoner (SelF-Reasoner) that assesses the entailment relationship between the question and the candidate reasoning chain. Then, we proceed with CoT reasoning when the reasoning chain demonstrates confidence; otherwise, we opt to predict the answer directly. SelF-Reasoner improves the fine-tuned T5 baseline consistently over the ScienceQA, ECQA, and LastLetter tasks. Code is available at `https://github.com/LibroWu/SelF-Reasoner`.

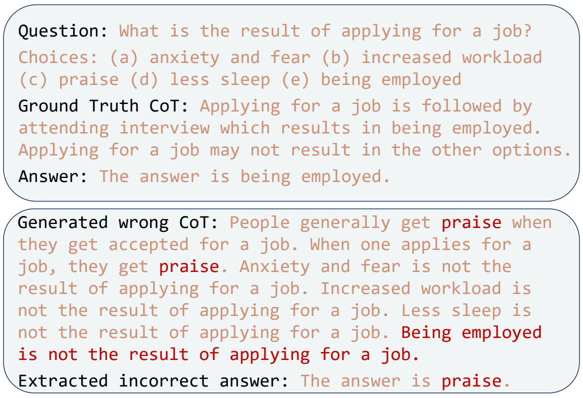
1. Introduction

> *"I will select their good qualities and follow them, their bad qualities and avoid them."*
>
> *Confucius* (551 BC - 479 BC)

Large language models (Brown et al., 2020; Thoppilan et al., 2022; Rae et al., 2021; Chowdhery et al., 2022, LLMs) have exhibited impressive capabilities in various reasoning tasks, including arithmetic and symbolic reasoning, by generating intermediate chain-of-thought (CoT) reasoning steps (Nye et al., 2022; Wei et al., 2022; Kojima et al., 2022).

Although CoT approaches have shown improvements in reasoning performance and interoperability, there are still two main challenges that hinder the widespread adoption of those approaches: (i) indecomposable questions, which refer to simple questions that cannot be decomposed into smaller sub-questions; (ii) erroneous reasoning chains, which involve mistakes in the logical and commonsense reasoning processes (Wei et al., 2022; Kojima et al., 2022; Zhang et al., 2022), and may even result in hallucinations (Wang et al., 2022a; Zhang et al., 2023) or unfaithful explanations (Turpin et al., 2023).

The above imperfect nature of CoT essentially leads to inferior results when using CoT instead of directly predicting answers, especially for small-scale language models prone to generating flawed reasoning chains (Zhang et al., 2023). Consequently, small-scale language models are unlikely to achieve better reasoning performance with self-generated CoT alone, unless distilling the effective CoT from LLMs to minimize the occurrence of erroneous reasoning chains (Li et al., 2022a; Magister et al., 2022; Ho et al., 2022).



Question: What is the result of applying for a job?
Choices: (a) anxiety and fear (b) increased workload (c) praise (d) less sleep (e) being employed
Ground Truth CoT: Applying for a job is followed by attending interview which results in being employed. Applying for a job may not result in the other options.
Answer: The answer is being employed.

Generated wrong CoT: People generally get praise when they get accepted for a job. When one applies for a job, they get praise. Anxiety and fear is not the result of applying for a job. Increased workload is not the result of applying for a job. Less sleep is not the result of applying for a job. Being employed is not the result of applying for a job.
Extracted incorrect answer: The answer is praise.

Figure 1: An example of an invalid CoT reasoning from ECQA (Aggarwal et al., 2021; Wang et al., 2023). The errors are highlighted in red. The generated CoT is wrong at the first step, and the error continues to the end. However, when altering to direct prediction, this one-step reasoning question is solved correctly.

Existing studies have proposed verification methods to improve the correctness of reasoning chains (Cobbe et al., 2021; Zhao et al., 2023; Diao et al., 2023; Sun et al., 2023; Lightman et al., 2023). However, they deal with all

---

[*]Corresponding authors. This paper was partially supported by Joint Research Project of Yangtze River Delta Science and Technology Innovation Community (No. 2022CSJGG1400) and Joint Funds of the National Natural Science Foundation of China (No. U21B2020).

LREC-COLING 2024, pages 11325–11340
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

the questions concerned by CoT in general, without selectively discarding irrelevant CoTs when necessary. Besides, their primary focus lies in refining the reasoning chains, neglecting the entailment relationship between the question and candidate reasoning chain. To bridge the gap, this work aims to tackle both challenges as described above simultaneously.

In this work, we propose a novel approach called the selective filtering reasoner (SelF-Reasoner) that assesses the entailment relationship between the question and the candidate reasoning chain. SelF-Reasoner proceeds with CoT reasoning when the reasoning chain demonstrates confidence; otherwise opting to predict the answer directly. Figure 1 shows a case when the generated CoT is wrong but the direct prediction can be correct. The SelF-Reasoner is composed of three key components: (i) a reasoner responsible for generating the candidate chain; (ii) an answerer module capable of predicting the final answer directly or extracting it from the question-solution pair; (iii) a CoT Filter designed to discard invalid reasoning chains and enhance the model's performance by utilizing the effective reasoning chain.

We implement SelF-Reasoner on the fine-tuned T5 models, revealing that small-scale language models can also benefit from CoT if equipped with our selective filtering mechanism. Experimental results on benchmarks show that our proposed method SelF-Reasoner improves the fine-tuned T5 baseline consistently over the ScienceQA(Lu et al., 2022a), ECQA, and LastLetter tasks. In summary, our main contributions are as follows:

(i) We proposed a selective filtering reasoner (SelF-Reasoner) to perform CoT only as necessary and mitigate the detrimental effects of erroneous reasoning chains.

(ii) Our SelF-Reasoner outperforms the fine-tuned CoT/vanilla baseline on ScienceQA, ECQA, and LastLetter datasets, advancing the effectiveness of CoT in small-scale language models.

(iii) We analyze the obstructions of fine-tuning CoT on language models and conclude common types in invalid generated CoT.

2. Related Work

Two lines of research are key to our work: chain of thought prompting and fine-tuning language models to reason.

2.1. Chain-of-thought Prompting

CoT prompting is a gradient-free strategy that encourages LLMs to generate the incremental thought processes required to reach a desirable outcome (Nye et al., 2022; Wei et al., 2022; Kojima et al., 2022). This approach aids LLMs in producing a chain of intermediate steps that leads to the final response to a question. According to the number of CoT examples in the prompt, we categorize CoT prompting into two primary methods: Zero-Shot CoT (Kojima et al., 2022) and Few-Shot CoT (Wei et al., 2022).

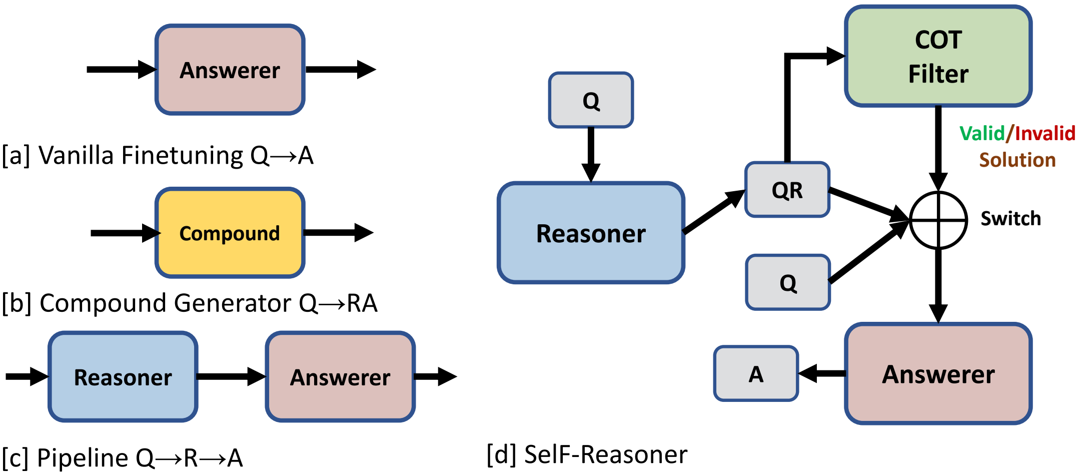
Zero-Shot CoT. Kojima et al. (2022) showed that LLMs are good zero-shot reasoners who already obtain the capability to reason from large-scale pretraining. A task-agnostic *magic spell* like *Let's think step by step* can make the LLMs generate the immediate reasoning steps and give the final answer. However, we do not find similar behavior in the smaller language models, mainly due to the limited training data and memory ability. Fine-tuning with a specific generation format further reduces the possibility of doing Zero-Shot CoT. So we fine-tune language models to generate reasoning chains instead of Zero-Shot CoT.

Few-Shot CoT. Few-Shot CoT achieves more robust performance by harnessing reasoning ability with the help of specially designed or self-created reasoning demonstrations. Researchers have been looking into various strategies to enhance the effectiveness of Few-Shot CoT. Least-to-most prompting (Zhou et al., 2022) decomposes complex problems into more manageable sub-problems to be solved sequentially. Wang et al. (2022c) introduced a self-consistency decoding strategy in that various reasoning trajectories are sampled to vote for the final answer. Cobbe et al. (2021) further trained verifiers on math problems to rank the candidate reasoning chains to get the most reliable reasoning chain. However, significant effort is still necessary to build the human-annotated reasoning chains in these methods. To overcome this limit, Zhang et al. (2022) proposed Auto-CoT utilizing Zero-Shot CoT and clustering to generate reasoning demonstrations automatically.

The performance of Few-Shot CoT can be influenced by the context and quality of the thought processes involved in the demonstrations. On certain specific tasks, utilizing fine-tuned language models may result in superior performance compared to using few-shot prompting LLMs.

2.2. Fine-tuning Language Models to Reason

Inspired by the success of CoT in LLMs, we investigate the possibility of applying CoT fine-tuning to smaller language models. Concurrent works (Li et al., 2022a; Magister et al., 2022; Ho et al., 2022; Li et al., 2022b; Wang et al., 2023)



[a] Vanilla Finetuning Q→A

[b] Compound Generator Q→RA

[c] Pipeline Q→R→A

[d] SelF-Reasoner

Figure 2: (a) Vanilla fine-tuning model that directly predicts the answer. (b) A compound generator that produces the reasoning chain and the answer simultaneously. (c) The two-level pipeline produces the reasoning chain first and then extracts the answer from it. (d) **Our SelF-Reasoner**. The Reasoner is fine-tuned to generate a solution based on the input question description. The Answerer is the model that can predict the answer directly or extract the answer from a question-solution pair. CoT Filter can determine whether the solution generated can be passed to the Answerer. Here, Q stands for Question, R stands for Reasoning Chain, and A stands for Answer. The symbol QR means concatenating Q and R, RA means concatenating R and A.

mainly utilize LLMs to create reliable reasoning chains, but these techniques still depend on LLMs. Hsieh et al. (2023) frames CoT learning as multi-task learning with rationales and shows their method can reduce the model size and the training data to achieve better performance than LLM. Wang et al. (2023) utilizes LLM to generate CoT and fine-tunes a smaller answerer under counterfactual training to make sounder use of the rationales. However, these works don't discard the produced invalid CoT even if it will lead to an incorrect answer. In our work, we focus on the fine-tuning process, and, different from Wang et al. (2023), we explicitly introduce a CoT filter to alleviate the effects of misleading CoTs.

Lu et al. (2022a) constructed a multimodal benchmark called Science Question Answering (ScienceQA) and fine-tuned language models to produce reasoning chains utilizing the annotated rationales in the dataset. They found that smaller language models, such as UnifiedQA (Khashabi et al., 2020), can benefit from CoT fine-tuning. However, in their setting, the model first provides the answer and then the reasoning chain, which only focuses on rationalization like WT5 (Raffel et al., 2019). In this paper, we thoroughly examine the CoT fine-tuning on this dataset and reflect on the provided reasoning chains.

3. Approach

This section will first present the details of our baseline for comparisons and our design of the SelF-Reasoner.

3.1. Baselines

We conduct experiments on three baselines: vanilla fine-tuning, compound generator, and two-level pipeline. For simplicity, let *Q* denote *Question*, *R* denote *Reasoning Chain*, and *A* denote *Answer*. Format *Q-R* means to input the question and generate the reasoning chain.

Vanilla Fine-tuning. In this setting (Figure 2 (a)), we fine-tune the model only to generate the final answer (format *Q-A*). Concretely, given the input question, the model generates the answer directly:

$$A = \textbf{Answerer}(Q). \tag{1}$$

Compound Generator. The compound generator (Figure 2 (b)) produces the reasoning chain and final answer simultaneously. Given the input question, the compound generator generates an output sequence with the reasoning chain and answer:

$$\{R, A\} = \textbf{Compound}(Q). \tag{2}$$

Depending on the positions of the reasoning chain and answer, we name the format where the reasoning chain is before the answer (*Q-RA*) as the CoT fine-tuning and the format where the reasoning chain is after the answer (*Q-AR*) as the reasoning chain fine-tuning. We will show that the practice of compound CoT fine-tuning is beset with the possibility of missing answers in the output (Section 4.4). To rectify this problem, we propose the implementation of a two-level pipeline, which will be presented subsequently.

Two-level Pipeline. In the two-level pipeline (Figure 2 (c)), there are a reasoner and an answer extractor. The reasoner produces only the reasoning chain and appends it to the question. Then the answer extractor extracts the final answer:

$$R = \textbf{Compound}(Q), \quad (3)$$

$$A = \textbf{Answerer}(Q \circ R). \quad (4)$$

The reasoner is fine-tuned in the format *Q-R*, and the extractor is fine-tuned in the format *QR-A*. We will show that the reasoner can provide misleading reasoning chains, leading to the answer extractor producing incorrect responses (Section 4.4 & 5.1). In light of this finding, a CoT filter that identifies invalid CoTs has been integrated into the system to form the SelF-Reasoner introduced in the following subsection.

3.2. SelF-Reasoner

The SelF-Reasoner (Figure 2 (d)) consists of a reasoner, an answerer, and a CoT filter. The reasoner is fine-tuned on the format *Q-R*:

$$R = \textbf{Compound}(Q). \quad (5)$$

The CoT filter will determine whether the reasoning chain is valid, thus filtering out incorrect reasoning chains. To train the CoT verifier, we use the generated CoT from the two-level pipeline to construct *QR-label* pair. If the generated CoT leads to the correct answer, then label this CoT as valid (label=1). Otherwise, the label is invalid (label=0):

$$v = \textbf{Verify}(Q \circ R). \quad (6)$$

The answerer is fine-tuned on the format *Q-A* augmented with the format *QR-A*, so it can directly predict the final answer or extract the answer from the question-CoT pair.

$$A = \begin{cases} \textbf{Answerer}(Q \circ R), & v = 1 \\ \textbf{Answerer}(Q), & v = 0. \end{cases} \quad (7)$$

To train the verifier, we use the T5 (Raffel et al., 2020) encoder extended with a linear classification layer as our verifier model. We load the convergent checkpoint of the answerer as the initial parameters before fine-tuning if possible. It can accelerate the training procedure and improve convergent accuracy.

We include more implementation details in Appendix A.3.

The verifier can also be logical rules to avoid extra computation. We use the rule that **"the given word should appear in the valid CoT"** to filter out invalid CoT in the LastLetter Task.

4. Experiments

This section will describe our experimental setup regarding datasets, language models, and evaluation, and present the main results.

4.1. Datasets

We use ScienceQA (Lu et al., 2022a), ECQA (Aggarwal et al., 2021; Talmor et al., 2019) and LastLetter for our experiments. ScienceQA is a benchmark that consists of multimodal multiple-choice questions with a diverse set of science topics and annotations of their answers with corresponding lectures and explanations. ECQA is a human-annotated version of CommonsenseQA (Talmor et al., 2019), which is a 5-choice QA dataset assessing general commonsense reasoning using concepts from ConceptNet (Speer et al., 2018). LastLetter is the task of concatenating the last letters of given words. More details about datasets can be found in Appendix A.1.

4.2. Language Models

We use UnifiedQA-T5 (Khashabi et al., 2020) as our primary language model to align with the experiments on the ScienceQA paper. Also, the imperfect nature of CoT is more salient in such language models. We select UnifiedQA-T5-small (60M), UnifiedQA-T5-base (220M), and UnifiedQA-T5-large (770M). More implementation details are included in Appendix A.2.

4.3. Evaluation

We use the accuracy of the answer (exactly matching for produced answer sentence) to measure the performance. To measure the quality of the generated reasoning chain, we use automatic metrics, such as BLEU-1/4 (Papineni et al., 2002), ROUGE-L (Papineni et al., 2002), and sentence similarity, as the settings on the ScienceQA paper. We use the Sentence-BERT network (Reimers and Gurevych, 2019) to encode the generated and ground truth reasoning chains and compute the cosine similarity as the metric Similarity.

Besides the automatic evaluations, we further conduct a human study to investigate our model performance more comprehensively. We randomly sample 50 instances from the test set and evaluate the entailment, completeness, and correctness. Entailment means the CoT entails the extracted answer. Completeness means that the CoT is complete. Correctness means that the CoT is correct and relevant to the question.

4.4. Main Results

In this section, we will present the main results and observations of our baselines, followed by a discussion of our SelF-Reasoner. Table 1 presents the main results of SelF-Reasoner compared with our baselines and existing methods of the ScienceQA benchmark. Tabel 3 presents the results of the ECQA dataset. Table 4 presents the results of the LastLetter task.

Vanilla Fine-tuning is a Strong Baseline. On ScienceQA benchmark, UnifiedQA$_{\text{Large}}$ achieves an accuracy of 86.53%, which is close to the human performance from Table 1, indicating that vanilla fine-tuning is a strong baseline.[1] Despite the high accuracy, vanilla fine-tuning still lacks interpretability. Then our next objective is to further elicit the model's reasoning capacity to explain its thought process while maintaining its accuracy.

Compound Generator Suffers from Imperfect CoT. The accuracy of CoT fine-tuning is lower than that of vanilla fine-tuning, in contrast to the benefits of CoT prompting in LLMs. The main reason is that the loss of the answer part is weakened by the CoT part. We also observe that the outputs are too lengthy in the setting of RA; thus, answers are not given due to the maximum length limits. We find that the ratios of missing answers in the training set are 0.8% in format *Q-EA* and 5.9% in format *Q-LEA*. Due to the answer missing issue, the performance of the *Q-RA* model is inferior to that of the *Q-AR* model, in agreement with the findings reported in Lu et al. (2022a). However, we do not adopt the AR format because it is more of a posterior rationalization than a CoT.

The Pipeline Method Narrows the Gap in Performance. The pipeline method narrows the gap between the compound generator and the vanilla fine-tuning (Table 1). We conduct a thorough evaluation of the produced reasoning chain in Section 5.1. By evaluating the respective cases, we find that some questions that the pipeline fails to answer can be properly solved by the vanilla fine-tuning model and vice versa. Vanilla fine-tuning and CoT fine-tuning can cause the model to acquire different segments of knowledge. This property can be utilized to augment the system's performance.

SelF-Reasoner Performs the Best. Our SelF-Reasoner gets the best performance consistently over the ScienceQA, ECQA, and LastLetter tasks.

On the ScienceQA benchmark (Table 1), the base and large SelF-Reasoner models guarantee significant improvement over the pipeline, and slightly outperform the strong vanilla fine-tuning model under both base and large sizes, respectively. Specifically, our best SelF-Reasoner model outperforms the previous state-of-the-art model (among the text-only models) Chameleon (GPT-4). Our SelF-Reasoner also significantly outperforms GPT-3 (CoT) by 12.07% and ChatGPT (CoT) by 8.93%, using a much smaller model size.

SelF-Reasoner also performs the best on ECQA. Table 3 presents the accuracy performance on the ECQA dataset. Though the T5-base model can not learn to produce CoT as well as human experts, the verifier alleviates the insufficiency and boosts the accuracy. SelF-Reasoner successfully meets our goal of creating a highly accurate and interpretable model. There is still a gap between the SelF-Reasoner and PINTO (61.67% as reported in Wang et al. (2023)) because PINTO gets LLMs involved to produce the rationale at test time, which can cause large computation or memory costs. In contrast, our approach does not rely on LLMs; thus it is more generally effective.

On the LastLetter task (Table 4), where the pipeline outperforms the vanilla fine-tuning, SelF-Reasoner can still benefit the performance. We apply the simple rule that **"the given word should appear in the valid CoT"** to filter out invalid CoTs because we find that the reasoner often replaces the given word due to randomness in sampling. For example, "speakers" can be replaced by "speaking". Without much overhead, the SelF-Reasoner outperforms the pipeline by 3.26%.

Overall, the results show that SelF-Reasoner successfully meets our goal of creating a highly accurate and interpretable model. The positive outcome indicates the significance of the CoT filter in fine-tuning language models. A detailed analysis of the pipeline component and the CoT filter has been conducted in Section 5.1 and Section 5.2, respectively.

5. Analysis

To understand how SelF-Reasoner works and gain insights, we analyze the generated reasoning

---

[1] There is a 13% discrepancy between the performance of vanilla fine-tuning reported by ours and Lu et al. (2022a). The discrepancy is discussed in Appendix A.4.

Method	Model	Learning	Format	Accuracy
Lu et al. (2022a)	Human	-	*Q-A*	88.4
	GPT-3 (CoT)	In-Context Learning	*Q-ALE*	75.17
	UnifiedQA$_{\text{Base}}$	fine-tuning	*Q-A*	70.12
	UnifiedQA$_{\text{Base}}$	fine-tuning	*Q-AE*	73.33
	UnifiedQA$_{\text{Base}}$	fine-tuning	*Q-ALE*	74.11
Lu et al. (2023)	ChatGPT (CoT)	In-Context Learning	*Q-EA*	78.31
	GPT-4 (CoT)	In-Context Learning	*Q-EA*	83.99
	Chameleon (ChatGPT)	In-Context Learning	*Q-EA*	79.93
	Chameleon (GPT-4)	In-Context Learning	*Q-EA*	86.54
Vanilla	UnifiedQA$_{\text{Samll}}$	fine-tuning	*Q-A*	71.54
	UnifiedQA$_{\text{Base}}$	fine-tuning	*Q-A*	83.09
	UnifiedQA$_{\text{Large}}$	fine-tuning	*Q-A*	86.53
Compound	UnifiedQA$_{\text{Base}}$	fine-tuning	*Q-ALE*	76.13
	UnifiedQA$_{\text{Base}}$	fine-tuning	*Q-EA*	77.71
	UnifiedQA$_{\text{Base}}$	fine-tuning	*Q-LEA*	73.97
Pipeline	UnifiedQA$_{\text{Small}}$	fine-tuning	*Q-E*→ *QE-A*	66.37
	UnifiedQA$_{\text{Base}}$	fine-tuning	*Q-E*→ *QE-A*	79.32
	UnifiedQA$_{\text{Large}}$	fine-tuning	*Q-E*→ *QE-A*	84.98
SelF-Reasoner	UnifiedQA$_{\text{Small}}$	fine-tuning	SelF-Reasoner	69.55
	UnifiedQA$_{\text{Base}}$	fine-tuning	SelF-Reasoner	83.45
	UnifiedQA$_{\text{Large}}$	fine-tuning	SelF-Reasoner	**87.24**

Table 1: Accuracy (%) of each baseline on test split. In the format part, *Q* = *Question*, *A* = *Answer*, *E* = *Explanation*, *L* = *Lecture*. We list the results from ScienceQA (Lu et al., 2022a), ChatGPT, GPT-4 (Lu et al., 2023) for comparison. *L* and *E* can be treated as reasoning chain. So LEA/EA and ALE/AE correspond to the standard RA and AR as defined in Section 3.1, respectively. Our SelF-Reasoner (Large) is comparable in accuracy to a human's.

Model	Split	BLEU-1	BLEU-4	ROUGE-L	Similarity	Complete	Entailment	Correct
Base	Lead to Correct Answer	0.914	0.776	0.910	0.937	1.00	1.00	0.94
	Lead to Incorrect Answer	0.789	0.660	0.797	0.860	1.00	1.00	0.02
	All	0.892	0.756	0.891	0.924	-	-	-
Large	Lead to Correct Answer	0.937	0.810	0.929	0.949	1.00	0.98	0.96
	Lead to Incorrect Answer	0.775	0.642	0.784	0.847	1.00	1.00	0.02
	All	0.917	0.788	0.910	0.936	-	-	-

Table 2: Automatic metrics (BLEU-1/4, ROUGE-L, Similarity) and human evaluation of generated explanations. We evaluate these metrics on different splits of the produced CoT according to whether they can lead to the correct answer. Details of human evaluation are shown in Appendix A.8.

Method	Vanilla	Pipeline	SelF-Reasoner
Accuracy	58.07	54.95	**58.48 (+3.5)**

Table 3: Accuracy (%) on test split of ECQA. The backbone model is UnifiedQA-base. SelF-Reasoner outperforms the pipeline by 3.5%.

Method	Vanilla	Pipeline	SelF-Reasoner
Accuracy	64.22	76.80	**80.06 (+3.26)**

Table 4: Accuracy (%) on test split of LastLetter. The backbone model is UnifiedQA-base. SelF-Reasoner outperforms the pipeline by 3.26%.

chains and the influence of the CoT filter. An incorrect CoT generated and filtered out by SelF-Reasoner is shown in Figure 1. More cases can be found in Appendix A.9, Table 9, Table 10, and Table 11 in Appendix.

5.1. Analysis on Generated Reasoning Chains

In this section, we evaluate the pipeline component of SelF-Reasoner by analyzing the produced reasoning chains. The evaluation includes both automatic metrics and human evaluation, and 50 examples from each data split are sampled for the human evaluation. The results show that the system makes some typical mistakes in the primary parts of the reasoning chains it produces. We also discuss the obstruction to generating perfect

Model	Vanilla	Pipeline	Random	SelF-Reasoner
Small	71.54	66.37	68.76	69.55
Base	83.09	79.32	81.61	83.45
Large	86.53	84.98	86.09	87.24

Table 5: Ablation on the CoT filter on ScienceQA benchmark. Random refers to randomly choosing vanilla fine-tuning and pipeline to produce the answer.

reasoning chains.

CoT fine-tuned model can produce invalid reasoning chains. In Table 2, BLEU and ROUGE metrics of correct samples are higher than the ones of incorrect samples, suggesting a quality gap in produced reasoning chains. The high Similarity metric (over 0.8) of invalid CoT indicates that the structure of the invalid CoT is similar to the ground truth. The human evaluation draws the same conclusion.

The case study illustrates that the key and necessary objects, described as "bridging objects" in Wang et al. (2022b), of the invalid reasoning chain mainly differ from the ground truth. We conclude some typical mistakes: (i) The bridging objects are missing; (ii) The bridging objects are mismatched; for example, feature A is attributed to object B, and feature B is attributed to object A; (iii) The bridging objects are wrong. These mistakes suggest the small language models are deficient in memorizing knowledge.

Incorrect reasoning chains can be traced back to both the model itself and the training data used. The model struggles to memorize all the necessary knowledge for the reasoning process, leading to errors in crucial parts of the reasoning chain.

Our analysis of the training data revealed that certain reasoning chains on the ScienceQA dataset do not precisely conform to the chain-of-thought format. More discussion can be found in Appendix A.7. Potential future work could focus on the impact of the reasoning chain's format on the CoT fine-tuning.

5.2. Influence of the CoT Filter

This section discusses the CoT filter component of SelF-Reasoner. The influence of the filter on the pipeline is evaluated, and the upper bound capabilities of SelF-Reasoner with an ideal filter are discussed.

The CoT Filter's Contribution to Better Accuracy Table 5 demonstrates the ablation on the CoT filter. SelF-Reasoner constantly outperforms the random baseline, indicating the significance of the filter.

Generator	Filter	Valid Acc	Invalid Acc	Acc	F1
Base	Base	76.96	76.39	76.84	0.841
	Large	81.30	81.64	81.37	0.874
Large	Base	74.97	75.03	74.98	0.836
	Large	80.07	78.17	79.78	0.871

Table 6: Accuracy and F1 score of the CoT filter on classifying the generated reasoning chain on ScienceQA benchmark. Valid/Invalid Acc refers to the filter's accuracy in discriminating valid/invalid reasoning chains. Acc is the overall accuracy.

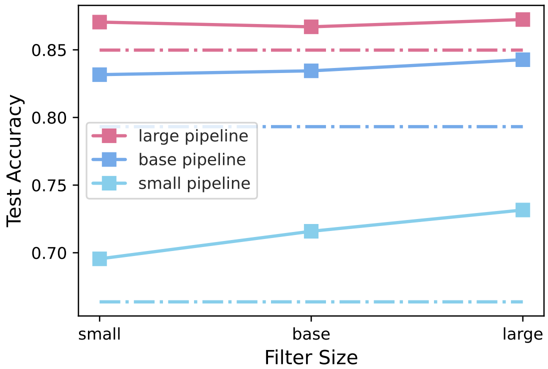


Figure 3: The "scaling law" in the size of the CoT filter and pipeline on ScienceQA benchmark. The dashed line presents the accuracy of the pipeline.

We further evaluate the prediction accuracy of the CoT filter for classifying the produced CoT from the base and large generators (Table 6). The filters demonstrate comparable capability in discriminating valid and invalid reasoning chains, though there is still potential for advancement. About 30% CoTs are filtered out by the filter, while the invalid CoT rate is about 25%.

"Scaling Law" of SelF-Reasoner Figure 3 shows the "scaling law" in the size of the CoT filter and pipeline. A CoT filter can consistently enhance the accuracy of pipelines. As the filter is made larger, the improvement can be more significant; however, augmenting the size of a pipeline often leads to diminishing returns in terms of improvement, suggesting that a larger generator can produce invalid reasoning chains that are more similar to valid ones, thereby making it harder for the filter to differentiate them.

We include the discussion on the upper bound of the Self-Reasoner in Appendix A.6.

6. Discussion

In the discussion section, we delve into several key aspects related to CoT fine-tuning and the challenges associated with optimizing the performance of SelF-Reasoner.

6.1. CoT Fine-tuning in a Specific Task

In this work, we observe that LLMs like GPT-3 may not consistently outperform fine-tuned small language models, even in tasks that involve generating complex and lengthy chain-of-thought sequences. This may be attributed to the fact that randomly chosen CoT prompts for LLMs may not effectively elicit optimal performance from LLMs, as the process of constructing such prompts can be both time-consuming and costly. Additionally, utilizing a small language model to achieve superior performance in a specific task is often a more straightforward approach regarding training and deployment.

6.2. The Role of the Reasoning Chain

Through the case study, we find that the small language model benefits from templated reasoning chains due to its limited capacity. Despite the success of our model on the three datasets, the specific role of the reasoning chain within the prediction procedure remains uncertain. To further investigate the impact of the reasoning chain, future research may employ adversarial techniques such as incorporating noise and altering key parts ("bridging objects") or the reasoning templates, as outlined in Wang et al. (2022b). This will aid in determining whether the improvement comes from the extra information introduced by the reasoning chain during the training phase or if the reasoning structure plays a significant role. Moreover, potential future work could focus on the impact of the reasoning chain's format on the CoT fine-tuning.

6.3. The Role of CoT Filter

The comprehensive evaluation of the reasoning chain is still dependent on human labor to examine individual sentences. Assessing the correctness of the reasoning chain, instead of merely determining if it leads to the correct answer, remains a complex task. The CoT filter enhances the accuracy and interpretability of the SelF-Reasoner, enabling the exclusion of misleading CoTs. However, the method of detecting invalid reasoning chains from valid ones is not currently understood. Future research can focus on developing interpretable filtering techniques utilizing rationalization methods and creating a toolkit for analyzing the quality of reasoning chains to aid in advancing CoT studies.

6.4. Obstructions on the Way to Perfect CoT

Small language models may encounter significant obstacles in generating both reasoning chains and answers in a single turn, owing to the restricted maximum input and output length. Furthermore, it can be difficult for these models to maintain coherence in longer output sequences. Due to their limited number of parameters, small language models may suffer from difficulties in fully internalizing the intricate relationships present within the training data. This can result in a weaker understanding of the relationship between questions and answers when the primary focus is on learning the structure and complex relationships within reasoning chains. Additionally, it should be noted that not all tokens within a reasoning chain hold equal importance. This can lead to the model utilizing its already insufficient parameters to retain less important tokens and potentially overlooking crucial information. To address these issues, future research can explore the implementation of joint training methods that incorporate both rationale loss and answer loss. Moreover, incorporating token rank information, as annotated within the reasoning chains of the training data, into the training process may improve the performance of fine-tuning CoT in language models.

6.5. Reflections on the Reasoning Chains from the Dataset

The reasoning chains in the training data play a crucial role in CoT fine-tuning. There exists a trade-off between templated reasoning chains and those of diverse forms. The use of templated forms allows for models to learn the reasoning skeleton more efficiently, enabling them to focus more on learning the knowledge and the relationship between the question and the answer. However, this approach results in a lack of diversity in the produced CoT and is therefore not suitable for implementing techniques such as self-consistency (Wang et al., 2022c) to further improve performance. On the other hand, utilizing more diverse training data poses a challenge for the model to learn effectively. Additionally, the annotation of reasoning chains for existing benchmarks requires a significant amount of effort. Human annotation is costly and time-consuming, and the reasoning chains annotated by humans may not adhere to the correct CoT format. One potential approach to replace human annotation is to utilize large language models to generate reasoning chains. However, the inference of LLMs is also costly, and their performance is not perfect. The CoT produced by large language models may still be incorrect and misleading, therefore requiring human verification.

7. Conclusion

This paper presents a method using fine-tuning to enhance the reasoning abilities of a language model. Additionally, we incorporate a CoT filter that can identify and exclude invalid CoTs to form SelF-Reasoner, thereby preventing invalid CoTs from negatively impacting the final answer prediction. The proposed SelF-Reasoner shows a significant performance improvement compared to two-level pipeline approaches, guaranteeing interpretability while maintaining accuracy. In addition, our approach has been shown generally effective across datasets, which achieves consistent performance gains on the ScienceQA, ECQA, and LastLetter datasets. We also conduct an in-depth analysis of the reasoning chains and training data to identify obstacles to achieving perfect CoT.

Limitations

Three limitations may be addressed in future studies. The first limitation is the increased model size over a single model because we need to train an additional filter. The second limitation is that the filter may still spare incorrect rationales, which can be improved by more effective negative sampling strategies in the filter training. The third limitation is that we evaluate our method in three datasets because most datasets lack effective annotated CoT for fine-tuning.

Ethical Considerations

The primary ethical concern related to this work is the potential for bias in the content generated by the fine-tuned language models used. However, because the focus of this work is the science or commonsense question-answering task, which relies on objective world knowledge and facts rather than subjective statements, the issue of bias is not as significant.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065, Online. Association for Computational Linguistics.

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

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*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single QA system. *CoRR*, abs/2005.00700.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022)*.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. 2022a. Explanations from large language models make small reasoners better.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. 2022b. Explanations from large language models make small reasoners better.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022a. Learn to explain: Multimodal reasoning via thought chains for science question answering. *arXiv preprint arXiv:2209.09513*.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022b. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Teaching small language models to reason.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2022. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2018. Conceptnet 5.5: An open multilingual graph of general knowledge.

Jiashuo Sun, Yi Luo, Yeyun Gong, Chen Lin, Yelong Shen, Jian Guo, and Nan Duan. 2023. Enhancing chain-of-thoughts prompting with iterative bootstrapping in large language models. *arXiv preprint arXiv:2304.11657*.

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.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. 2023. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *arXiv preprint arXiv:2305.04388*.

Boshi Wang, Xiang Deng, and Huan Sun. 2022a. Iteratively prompt pre-trained language models for chain of thought. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2714–2730.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022b. Towards understanding chain-of-thought prompting: An empirical study of what matters.

Peifeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2023. Pinto: Faithful language reasoning using prompt-generated rationales.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022c. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022)*.

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

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multi-modal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. *arXiv preprint arXiv:2305.03268*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A. Experiment Details

A.1. Dataset

We provide more details on the datasets used in our experiments.

ScienceQA (Lu et al., 2022a) is a multimodal dataset annotated with reasoning chains by human experts. It is available at https://scienceqa.github.io/. On ScienceQA, a data example consists of multimodal question-answering information annotated with background lecture and explanation. We use the captions provided in the dataset, which are generated by the model based on ViT (Dosovitskiy et al., 2020) and GPT-2 (Radford et al., 2019) to replace the visual context following (Lu et al., 2022b). The task is formulated as a text-to-text problem where the input Q is a concatenation of question, context, and options and the output is the reasoning chain R or answer A depending on the role of the module as defined in Section 3.

ECQA is a human-annotated version of CommonsenseQA (Talmor et al., 2019). The original CommonsenseQA is available at https://www.tau-nlp.sites.tau.ac.il/commonsenseqa. ECQA version is available at https://github.com/dair-iitd/ECQA-Dataset. PINTO (Wang et al., 2023) also provides a version annotated by LLM.

LastLetter is a task to concatenate the last letter of the given words. We constructed the dataset from the Google-10000-English repository. We sample 10,000 groups for training and 5,000 for tests (averagely split ranging from 1 to 5 words). The test words are not seen in training words.

We provide the dataset statistics in Table 7.

Table 7: Dataset statistics used in our experiments.

Dataset	Train	Validation	Test
ScienceQA	12726	4241	4241
ECQA	8520	1221	1221
LastLetter	10000	5000	5000

A.2. Implementation Details

We fine-tune the UnifiedQA for 20 epochs with a learning rate of 4e-5 and a weight decay of 0.01. We use batches of 4. We trained the base and large SelF-Reasoners on Nvidia RTX 2080Ti for 20 and 60 hours, respectively. The reported results are from models trained for 20 epochs. The maximum input sequence length is 512.

A.3. Verifier Training Details

To construct the training data of the verifier, we run the pipeline model on the training data. The instances with reasoning chain leading to correct predictions will be labeled as positive and those leading to incorrect predictions will be labeled as negative. To balance the number of positive and negative instances, we use a subpar reasoner which scores less precise in the training set to generate invalid instances.[2]

A.4. Discussion on Different Vanilla Fine-tuning Result Compared to Lu et al. (2022a)

The difference in the training setting can contribute to the discrepancy. Although we use the same training batch size and learning rate as Lu et al. (2022a), there are still variances in the details. We have a rating decay of 0.01, and we predict the answer directly instead of the choice. Furthermore, the difference in training time and convergence state also influences performance. Our result is double-checked by rerunning the experiments.

A.5. Confusion Matrice

Reasoner	Method Used	Correct Otherwise	Both Fail
Base	Directly Predict	154	358
	Extract	71	119
Large	Directly Predict	124	277
	Extract	38	102

Table 8: Confusion matrices of the SelF-Reasoner on the incorrect cases. Method Used refers to the adopted method to predict the answer. Correct Otherwise means if the other method were used, the question could be solved. Both Fail means neither method can solve the question.

A.6. Upper Bound of the SelF-Reasoner

The confusion matrices in Table 8 demonstrate the errors made by SelF-Reasoner. There is a certain amount of complex questions that can not be solved using both methods, indicating that the upper bound of SelF-Reasoner with an ideal filter is approximately 89% for base and 91% for large.

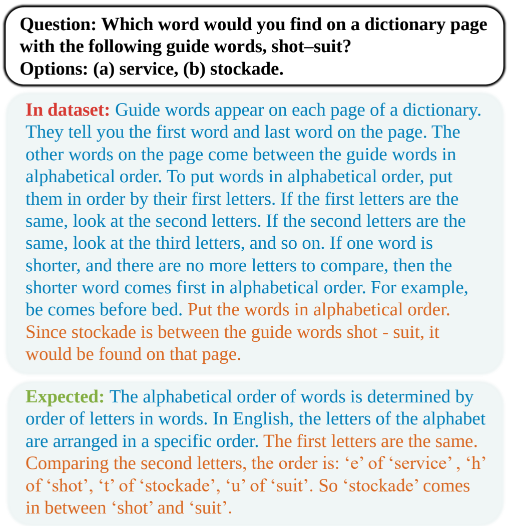
A.7. Incorrect CoT Format

One typical example is sorting words in alphabetical order, as seen in Figure 4. The background lecture about alphabetical order is too extensive for the model to process, and the solution part is overly simplistic, resembling a fill-in-the-blank task rather than a logical step-by-step problem-solving

---

[2] In ScienceQA, the convergent two-level pipeline can reach a score of 90% in the training set. So we use a subpar reasoner which scores 60% in the training set to generate invalid instances.

process. An example of an expected reasoning chain is also provided in Figure 4. Additionally, we find that some solutions present the answer first, followed by the explanation.



Question: Which word would you find on a dictionary page with the following guide words, shot–suit?
Options: (a) service, (b) stockade.

In dataset: Guide words appear on each page of a dictionary. They tell you the first word and last word on the page. The other words on the page come between the guide words in alphabetical order. To put words in alphabetical order, put them in order by their first letters. If the first letters are the same, look at the second letters. If the second letters are the same, look at the third letters, and so on. If one word is shorter, and there are no more letters to compare, then the shorter word comes first in alphabetical order. For example, be comes before bed. Put the words in alphabetical order. Since stockade is between the guide words shot - suit, it would be found on that page.

Expected: The alphabetical order of words is determined by order of letters in words. In English, the letters of the alphabet are arranged in a specific order. The first letters are the same. Comparing the second letters, the order is: 'e' of 'service' , 'h' of 'shot', 't' of 'stockade', 'u' of 'suit'. So 'stockade' comes in between 'shot' and 'suit'.

Figure 4: An example that does not conform to the CoT format and an expected reasoning chain. Blue texts are the knowledge lectures and Orange texts are the solutions.

A.8. Human Evaluation

In the human evaluation process, the specifications of the three metrics are as follows. **Complete** means the model generates a complete CoT (No early stop or repeating). **Entailment** means the extracted answer entails the CoT. **Correctness** means the reasoning chain is correct and relevant to the question (No irrelevant entities or verbs). 50 examples from each data split are sampled randomly for human evaluation.

A.9. Case Study

Following is a case from ECQA. The model generates a half-correct CoT. However, the answerer is misled by the generated CoT and gives an incorrect prediction. However, in SelF-Reasoner, the generated CoT is filtered and the answerer gives a correct prediction based on the question.

> question: Where can you put a picture frame when it's not hung vertically? choices: (A) wall. (B) newspaper (C) table (D) Car
>
> generated CoT: **Picture frames are hung vertically on the wall.(Correct part)** Picture frames are not hung vertically on the art show. **Newspaper or table is not a place where a picture frame can be put.(Misleading part)** Car is not a place where a picture frame can be put.
>
> ground truth CoT: Pictures frames are usually either hung on a wall or kept on a flat surface. A table has a flat surface. Art show usually does not have pictures that are not hung on the wall. Wall is a vertical surface. It is not possible to avoid hanging a picture vertically on a wall. Newspaper is an irrelevant option. Pictures are not generally hung in a car.
>
> prediction: Answer: The answer is wall.
> reference: Answer: The answer is table.

More cases are shown in Table 9, Table 10, and Table 11.

Table 9: Demonstrations from ScienceQA.

Q: Which of these organisms contains matter that was once part of the lichen? Choices: (A) bilberry (B) mushroom.

Predicted rationale: Use the arrows to follow how matter moves through this food web. For each answer choice, try to find a path of arrows that starts from the lichen. There are two arrows pointing to the mushroom. One arrow starts from the bear sedge, and the other arrow starts from the lichen. Neither the bear sedge nor the lichen has an arrow pointing to it. So, in this food web, matter does not move from the lichen to the mushroom.

Ground truth rationale: Use the arrows to follow how matter moves through this food web. For each answer choice, try to find a path of arrows that starts from the lichen. No arrow points to the bilberry. So, in this food web, matter does not move from the lichen to the bilberry.

A: The answer is (B) mushroom.

Q: Think about the magnetic force between the magnets in each pair. Which of the following statements is true? Choices: (A) The magnitude of the magnetic force is the same in both pairs. (B) The magnitude of the magnetic force is smaller in Pair 1. (C) The magnitude of the magnetic force is smaller in Pair 2.

Predicted rationale: The magnets in Pair 2 attract. The magnets in Pair 1 repel. But whether the magnets attract or repel affects only the direction of the magnetic force. It does not affect the magnitude of the magnetic force. Magnet sizes affect the magnitude of the magnetic force. Imagine magnets that are the same shape and made of the same material. The smaller the magnets, the smaller the magnitude of the magnetic force between them. Magnet A is the same size in both pairs. But Magnet B is smaller in Pair 2 than in Pair 1. So, the magnitude of the magnetic force is smaller in Pair 2 than in Pair 1.

Ground truth rationale: Magnet sizes affect the magnitude of the magnetic force. Imagine magnets that are the same shape and made of the same material. The smaller the magnets, the smaller the magnitude of the magnetic force between them. Magnet A is the same size in both pairs. But Magnet B is smaller in Pair 2 than in Pair 1. So, the magnitude of the magnetic force is smaller in Pair 2 than in Pair 1.

A: The answer is (C) The magnitude of the magnetic force is smaller in Pair 2.

Q: Which of the following contains a vague pronoun reference? Choices: (A) Abby was surprised when the article said that at least one-third of American adults get less than seven hours of sleep each night. (B) Abby was surprised when it said that at least one-third of American adults get less than seven hours of sleep each night.

Predicted rationale: The second answer choice contains a vague pronoun reference. The pronoun it could refer to the article or the night. The first answer choice shows a possible correction for the vague pronoun reference. It has been replaced with the article. Abby was surprised when the article said that at least one-third of American adults get less than seven hours of sleep each night.

Ground truth rationale: The first answer choice contains a vague pronoun reference. The pronoun it is used without its antecedent. The second answer choice shows a possible correction for the vague pronoun reference. It has been replaced with the article. Abby was surprised when the article said that at least one-third of American adults get less than seven hours of sleep each night.

A: The answer is (B) Abby was surprised when it said that at least one-third of American adults get less than seven hours of sleep each night.

Table 10: Demonstrations from ECQA.

Q: Reading newspaper one of many ways to practice your what? Choices: (A) literacy (B) knowing how to read (C) money (D) buying (E) money bank.

Predicted rationale: Reading newspaper is one of many ways to practice your literacy. Knowing how to read is not practiced. Money is not practiced.Buying is not practiced. Money bank is not a way to practice.

Ground truth rationale: Literacy means competency. Reading newspaper provides daily new information to oneself. One can maintain literacy with reading different information daily from newspaper. One cannot read newspaper without knowing how to read. Reading newspaper does not help in achieving other options.

A: The answer is (A) literacy.

Q: If you are prone to postpone work what will you have to do in order to finish on time? Choices: (A) eat (B) hasten (C) antedate (D) bring forward (E) advance.

Predicted rationale: If you are prone to postpone work, you will have to have to have a hasten in order to finish on time. Eat is not true as it is not necessary to finish on time. Antidate is not true as it is not necessary to finish on time. Bring forward is not true as it is not necessary to finish on time. Advance is not true as it is not necessary to finish on time.

Ground truth rationale: If you have to complete the work on time then you have to speed-up. Antedating will do work early than required, but not on time. Eating, bringing forward and working in advance are also not helpful to do work on time.

A: The answer is (B) hasten.

Q: Aside from water and nourishment what does your dog need? Choices: (A) bone (B) charm (C) petted (D) lots of attention (E) walked.

Predicted rationale: Bone is a solid surface of the body that is needed for growth, development, and development. Bone is a part of the body and is not needed for growth, development, or development. Bone is the part of the body and not the body part. Charm is a charm and not the body part. Petted is a human pet and not a body part. Lots of attention is the amount of time and attention given to a dog and not the body part. Walk is to walk and not the body part.

Ground truth rationale: Lots of attention is a special care or notice taken of someone or something, the regarding of someone or something as interesting or important. Aside from water and nourishment, our dog needs lots of attention. Bone is a treat that our dog will enjoy, and bone is nourishment and not what our dog needs the most. Charm is a quality of fascinating others and our dog doesn2̌019t have to fascinate others. Petted is to show affection and not a care that a dog requires. Walked is moved at a regular pace and that a dog can do by his own and not what he needs from you.

A: The answer is (D) lots of attention.

Q: Where can you put a picture frame when it's not hung vertically? Choices: (A) art show (B) wall (C) newspaper (D) car (E) table.

Predicted rationale: Picture frames are hung vertically on the wall. Picture frames are not hung vertically on the art show. Newspaper or table is not a place where a picture frame can be put. Car is not a place where a picture frame can be put.

Ground truth rationale: Pictures frames are usually either hung on a wall or kept on a flat surface. A table has a flat surface. Art show usually does not have pictures that are not hung on the wall. Wall is a vertical surface. It is not possible to avoid hanging a picture vertically on a wall. Newspaper is an irrelevant option. Pictures are not generally hung in a car.

A: The answer is (E) table.

Table 11: Demonstrations from LastLetter Task.

Q: immigrants, editorials, hierarchy

Predicted rationale: The last letter of the first word 'immigrants' is's'. The last letter of the second word 'editorials' is 's'. The last letter of the third word 'hierarchy' is 'y'.

Extracted Answer: The answer is ssy. (Correct)
Directly predicted Answer: The answer is ssy. (Correct)
Ground Truth Answer: The answer is ssy.

Q: poison, dame, cornell

Predicted rationale: The last letter of the first word 'poison' is 'n'. The last letter of the second word 'dame' is 'e'. The last letter of the third word 'cornell' is 'l'.

Extracted Answer: The answer is nel. (Correct)
Directly predicted Answer: The answer is nell. (Inorrect)
Ground Truth Answer: The answer is nel.

Q: departmental, dome, pressed, fascinating

Predicted rationale: The last letter of the first word 'departmental' is 'l'. The last letter of the second word 'dome' is 'e'. The last letter of the third word 'pressed' is 'd'. The last letter of the forth word 'funny' is 'y' .

Extracted Answer: The answer is ledy. (Incorrect, the rationale part is wrong)
Directly predicted Answer: The answer is ledg. (Correct)
Ground Truth Answer: The answer is ledg.

Q: bahrain, joyce, calibration, ambien

Predicted rationale: The last letter of the first word 'bahrain' is 'n'. The last letter of the second word 'joyce' is 'e'. The last letter of the third word 'calendar' is 'r'. The last letter of the forth word 'ambien' is 'n'.

Extracted Answer: The answer is nern. (Incorrect, extraction part is wrong)
Directly predicted Answer: The answer is nenn. (Inorrect)
Ground Truth Answer: The answer is ssy.