

Can Small Language Models Help Large Language Models Reason Better?: *LM-Guided Chain-of-Thought*

Jooyoung Lee^{1*}, Fan Yang², Thanh Tran², Qian Hu²
Emre Barut², Kai-Wei Chang², Chengwei Su²

Penn State University, PA, USA¹

Amazon AGI, MA, USA²

jfl5838@psu.edu, {fyaamz, tdt, huqia, ebarut, kaiwec, chengwes}@amazon.com

Abstract

We introduce a novel framework, **LM-Guided CoT**, that leverages a lightweight (*i.e.*, <1B) LM for guiding a black-box large (*i.e.*, >10B) LM in reasoning tasks. Specifically, the lightweight LM first generates a rationale for each input instance. The Frozen large LM is then prompted to predict a task output based on the rationale generated by the lightweight LM. Our approach is resource-efficient in the sense that it only requires training the lightweight LM. We optimize the model through 1) knowledge distillation and 2) reinforcement learning from rationale-oriented and task-oriented reward signals. We assess our method with multi-hop extractive question answering (QA) benchmarks, HotpotQA and 2WikiMultiHopQA. Experimental results show that our approach outperforms all baselines regarding answer prediction accuracy. We also find that reinforcement learning helps the model to produce higher-quality rationales with improved QA performance.

Keywords: Chain-of-Thought Prompting, Large Language Model, Reinforcement Learning, Knowledge Distillation

1. Introduction

Chain-of-Thought (CoT) prompting (Wei et al., 2022b) has gained attention as a means to elicit LMs’ reasoning abilities. By prompting the models with “Let’s think step by step” following the actual task description, the models first produce intermediate reasoning steps and then predict a task output. It has been shown to enhance the downstream task performance in complex reasoning domains such as arithmetic (Lewkowycz et al., 2022), commonsense (Jung et al., 2022), and symbolic (Khot et al., 2022) reasoning. However, there are several limitations to conventional CoT prompting. Firstly, the performance gains when compared to standard prompting (*i.e.*, without “Let’s think step by step”) are only likely to emerge in very large LMs, preferably 100+ billion parameters (Wei et al., 2022a). Moreover, the models may still generate low-quality rationales that are repetitive and vacuous (Ye and Durrett, 2022). This can be attributed to their tendencies to lack faithfulness to an input instance (Lanham et al., 2023) and to produce unaligned rationales and answers (Wang et al., 2022b; Turpin et al., 2023).

Given that CoT prompting is primarily effective with large language models, rectifying these undesirable behaviors through direct model modifications is non-trivial, particularly with constrained computational resources. Hence, we propose **LM-guided CoT**, a novel framework that leverages two

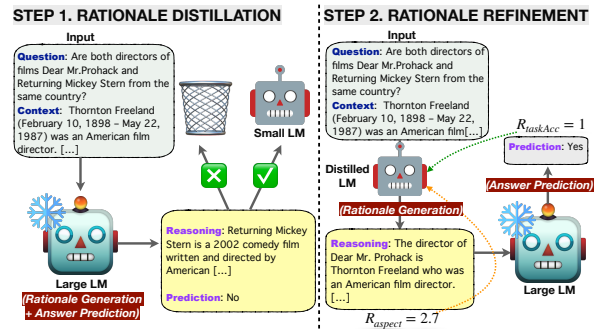


Figure 1: Illustration of our proposed method.

independent LMs (*i.e.*, a small LM for rationale generation and a large LM for answer prediction) for CoT reasoning. As shown in Figure 1, we first employ a vanilla knowledge distillation (KD) technique to the small LM with rationales generated by the large LM (§3.1). This initial step helps narrow the gap in reasoning capabilities between the smaller and larger LMs to a certain extent. To further improve the quality of rationales generated by the knowledge-distilled LM, we establish fine-grained measurements concerning 8 rationale-specific aspects (relevance, actuality, logicity, consistency, coherence, fluency, naturalness, and readability) and use them to optimize the knowledge-distilled LM with reinforcement learning (RL) (§3.2).

We conduct experiments on an extractive multi-hop question answering (QA) task using two popular benchmarks, HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020). Although our framework can be flexibly applied to a wide

*This work was done during her internship at Amazon.

range of LMs, we use FLAN-T5 (Longpre et al., 2023) models in this work because they are open-source and instruction-tuned on both QA and CoT data, enabling us to use it off-the-shelf without additional training or prompt engineering.¹ Our experiment results show that LM-guided CoT prompting outperforms both the standard prompting and the original CoT prompting. More precisely, we find that (1) LM-guided CoT with KD and self-consistency (SC) decoding strategy (Wang et al., 2022b) maximizes the performance gain; (2) RL contributes to a slight increase in overall rationale quality and task performance; (3) choosing the highest-quality rationales for the large LM does not always guarantee improved task performance. Overall, this approach provides an alternative to the direct optimization of the large LM by focusing on fine-tuning the comparatively smaller LM. Moreover, the clear separation of two fundamental sub-tasks within CoT reasoning grants practitioners greater control over each task.

2. Related Work

Rationale Distillation. For computation efficiency or task performance, recent literature has explored ways to improve small LMs’ reasoning abilities. Li et al. (2023); Shridhar et al. (2023); Ma et al. (2023) have experimented with rationale distillation, where a small student LM learns from a large teacher LM to generate CoT rationales. While these studies have mainly concentrated on comparing its performance in downstream tasks against that of large LMs, there has been limited investigation into addressing errors in the generated rationales that might have been inherited from the teacher model.

Rationale Evaluation and Refinement. In contexts beyond rationale distillation, there have been growing efforts to unravel which aspects of the generated reasoning steps contribute to its performance. Wang et al. (2022a) report that rationales’ logicity and relevance to the query are key factors in successful CoT reasoning. Few studies have measured the validity of reasoning steps from the lens of more diverse aspects like informativeness, coherence, and repetition, etc (Golovneva et al., 2022; Prasad et al., 2023). While RL has gained popularity as an approach for addressing misaligned behaviors in LMs, the field of rationale correction has seen limited research.

3. LM-guided Chain-of-Thought

The proposed framework consists of two LMs: a lightweight model M^S that focuses on generating

¹We also ran a small experiment with the [instruction-tuned Open LLaMa \(7B\) model](#) and confirmed that its

the optimal rationale given an input instance, and a black-box large model M^L that predicts an output based on the rationale generated by M^S .

3.1. Rationale Distillation

Rationale Generation. In general, multi-hop extractive QA datasets contain a list of questions Q , contexts C , and corresponding ground truth answers A . For each input (q, c) -output (a) pair, we need a corresponding ground truth rationale r to train M^S for rationale generation in a supervised manner. However, most of QA benchmarks do not provide r . Given that manual annotation for r is labor-extensive and time-consuming, we perform the vanilla KD technique, where we automatically obtain rationales (denoted as r') from a teacher model (*i.e.*, M^L) and use it to train a student model (*i.e.*, M^S). The prompt used for rationale generation from M^L is outlined as below:

Based on the provided context, answer the following question (Q) by reasoning step-by-step.
Context: c
Q: q
A : Let’s think step by step.

For generation, we use greedy decoding; *i.e.*, choosing the most plausible token at each generation step.

Rationale Filtering and Training. When it comes to knowledge distillation, data cleaning processes play a crucial role in preventing errors or noises included in the generation from the teacher LM being inherited to the student LM. Thus, we filter samples associated with unfaithful responses to the prompt (*i.e.*, not providing rationales prior to providing a final answer) and inaccurate answer prediction. Finally, we instruction-tune M^S using the following prompt:

Given a question (Q) and a context, generate a chain of reasoning step by step to answer the question.
Context: c
Q: q
Reasoning: r'

For the rest of the paper, we denote the rationale-distilled model as M^* .

3.2. Rationale Refinement

Annotation for Rationale Quality Measurement. Inspired by previous text and rationale generation evaluation metrics (Golovneva et al., 2022; Fu et al.,

performance is significantly worse than FLAN-T5 in a zero-shot setting.

Aspects	Descriptions
Factuality	Percentage (0.0-1.0) measuring if the reasoning is grounded based on the context (<i>Input: c & r'</i>)
Relevance	Percentage (0.0-1.0) measuring if the reasoning is relevant to the question (<i>Input: q & r'</i>)
Logicity	Binary (0 or 1) measuring if the reasoning is logical and can reach a final answer
Consistency	Binary (0 or 1) measuring if the reasoning remains consistent and coherent (<i>Input: q & r'</i>)
Coherence	Binary (0 or 1) measuring if the reasoning is without redundant information (<i>Input: q & r'</i>)
Fluency	Binary (0 or 1) measuring if the reasoning is well-written and grammatically correct (<i>Input: r'</i>)
Naturalness	Binary (0 or 1) measuring if the reasoning is natural and human-like (<i>Input: r'</i>)
Readability	Binary (0 or 1) measuring if the reasoning is easy to follow and understandable (<i>Input: r'</i>)

Table 1: Descriptions of 8 rationale aspects used for evaluation. q , c , and r' denote a question, context, and a corresponding rationale generated by the small LM, respectively.

2023), we attempt to quantify 8 linguistic aspects (factuality, relevance, logicity, consistency, coherence, fluency, naturalness, readability) of rationales generated by M^* in §3.1. Let us denote r^* as M^* -generated reasoning. Since there is no ground truth rationale r available for comparison, our metrics are reference-free, utilizing a pair of r^* and one of existing inputs (q or c). Table 1 describes each aspect type and input combinations. As we intend to use these metrics for reward scoring in RL, it is critical to have an accurate metric. Hence, we obtain a small set ($n=100$) of gold labels for all aspect types through human annotation. A detailed description of the annotation process and results is reported in §11.1.

Automatic Measurement for Rationale Quality.

Manual annotation offers substantial value, but within the context of RL, it becomes notably challenging due to frequent reward scoring. Therefore, we probe several ways to automate this process. Given that Ye and Durrett (2022) introduced the simple yet effective approach for assessing factuality and relevance through token-level lexical overlap, we choose to adopt their method. For the remaining 6 categories, two methods are considered. The first approach is to harness a large LM to function as reference-free NLG evaluators, inspired by recent works (e.g., Liu et al. (2023), Wang et al. (2023)). The second approach, in contrast, involves training a simple machine learning classifier using human-annotated data. Two approaches are described in depth in §11.2. Due to inference time efficiency and higher alignment scores to human annotators (see Table 5), we use the second method for all our experiments.

RL for Rationale Refinement. We next detail how to utilize established evaluation metrics as reward signals to update M^* in §3.1 with Proximal Policy Optimization (PPO) (Schulman et al., 2017). Given each input (q, c)-output (a) pair from training data, we first prompt M^* to generate a corresponding rationale r^* . During the generation process, an aspect-specific reward (denoted as R_{aspect}) is measured by summing all values returned from au-

tomatic evaluation metrics.² We then pass r^* to M^L to retrieve the answer prediction a^* and compute a task-specific reward (denoted as $R_{taskAcc}$). Specifically, we leverage the F1 score between the predicted answer and the ground truth answer:

$$R_{taskAcc} = \begin{cases} 1 & \text{if } F1(a, a^*) > 0.5, \\ 0 & \text{else.} \end{cases}$$

A final reward score for model training is $R_{aspect} + R_{taskAcc}$. Following Stiennon et al. (2020), we also incorporate penalties based on the Kullback Leibler (KL) divergence between the learned policy LM and M^S .

4. Experiments and Results

4.1. Experimental Setup

Model and Dataset. we utilize FLAN-T5 small (80M) for M^S and FLAN-T5 XXL (11B) for M^L . Both HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020) consist of an input question, an answer, along with 9-10 context paragraphs with supportiveness labels indicating whether the paragraph contains supporting facts. Due to the input token size limitation of FLAN-T5, we only use supporting paragraphs as context.

Training & Evaluation Setup. We use a randomly sampled subset of training data from two datasets (15K samples per data) for model training. After filtering unqualified reasoning, it results in 23K samples. All training-related hyperparameters can be found in §11.3. According to our preliminary experiments, the impact of CoT prompting appeared to be diminished due to two potential factors: (1) questions being overly simplistic, obscuring the significance of intermediate reasoning processes; (2) LMs already possessing pertinent background information (e.g. Flan-T5 is fine-tune on various question-answering datasets). To prevent models from answering based on parametric mem-

²We also test with normalization or weighted summation for the scores, but they did not affect the performance.

Prompt	Rationale Provision?	HotpotQA			2WikiMultiHopQA		
		EM	F1	Answer Inclusion	EM	F1	Answer Inclusion
standard prompting	✗	0.5	0.714	0.583	0.5	0.625	0.647
CoT prompting	✓	0.483	0.686	0.611	0.4	0.532	0.561
CoT prompting + SC	✗	0.503	0.70	0.624	0.471	0.603	0.625
LM-guided CoT prompting (KD)	✓	0.507	0.702	0.625	0.506	0.626	0.661
LM-guided CoT prompting (KD + SC)	✗	0.513	0.714	0.635	0.524	0.644	0.679
LM-guided CoT prompting (KD + R_{aspect})	✓	0.503	0.698	0.625	<u>0.507</u>	<u>0.631</u>	<u>0.665</u>
LM-guided CoT prompting (KD + R_{aspect} + $R_{taskAcc}$)	✓	<u>0.508</u>	<u>0.704</u>	<u>0.627</u>	0.503	0.622	0.657
LM-guided CoT prompting (KD + R_{aspect} + ranking)	✓	0.5	0.698	0.623	0.501	0.619	0.653

Table 2: Answer prediction performance results of baselines and our approach. We regard SC decoding as a non-rationale provision, given that this method can result in multiple variations of rationales, rather than a single one. Values in bold represent the highest scores and underlined values are the second highest scores.

Type	Description	Template
Standard prompting	Directly predicting the answer based on input	Based on the provided context, answer the following question (Q). Context: c Q: q A:
CoT prompting	Predicting the answer after generating the reasoning	Based on the provided context, answer the following question (Q) by reasoning step-by-step. Context: c Q: q A : Let’s think step by step.
LM-guided CoT prompting (<i>our method</i>)	Predicting the answer with conditional generation upon the LM-generated reasoning	Based on the provided context, answer the following question (Q) by reasoning step-by-step. Context: c Q: q A : Let’s think step by step. r' . Hence, the answer is

Table 3: Descriptions and templates of each prompt used for the answer prediction task. q , c , and r' denote a question, context, and a corresponding rationale generated by the small LM, respectively.

ory, we attempt to make the existing evaluation data more challenging, similar to the approaches taken by Ye and Durrett (2022) and Zhao et al. (2023). For each dataset, we leverage the prediction outcomes of standard prompting and select 1000 input instances that M^L answered correctly and an additional 1000 from questions where the model provided incorrect responses. This results in a total of 2000 samples for evaluation.

Baselines and Evaluation Metrics. We use standard prompting and CoT prompting as our baselines (see Table 3). We also experiment with the SC decoding strategy, where it samples multiple reasoning paths ($n=10$) and selects the most consistent answer. For evaluation, we report three metrics for the answer prediction task: (1) exact match (EM), computing whether the prediction exactly matches the ground truth answer, (2) F1, computing the average word overlap between the prediction and ground truth answer, and (3) answer

inclusion³, computing whether the ground truth answer is mentioned in the prediction.

4.2. Results

Baseline performance. As shown in Table 2, we find that M^L (equivalent to FLAN-T5 XXL) does not benefit from the original CoT prompting, as its performance dropped in both datasets (except for answer inclusion score for HotpotQA) when compared to standard prompting. This is consistent with prior works’ findings that models with less than 50B parameters are not good at reasoning. We also observe that the performance drop is more significant with 2wikiMultihopQA (nearly 10% for

³We included this measurement because models often provide more extensive responses (e.g., ground truth : World’s Best Goalkeeper → generation: IFFHS World’s Best Goalkeeper). This may be explained by our usage of contextual paragraphs as input.

EM and F1) than HotpotQA. Based on our manual inspection of incorrect predictions, M^L was prone to repeat sentences in the context and fail to provide a final answer to the questions. This hints that, when context gets too long, models face difficulties in digesting the content and establishing valid reasoning steps. Overall, self-consistency CoT prompting enables the model to noticeably recover from answer prediction errors, especially for 2WikiMultiHopQA.

LM-guided CoT performance. Table 2 shows a comprehensive breakdown of our method’s performance. Additionally, we explore an extension of our approach, which involves sampling multiple reasoning paths and subsequently ranking the most optimal rationales. Our method with only KD outperforms the original CoT prompting with 2% gain for HotpotQA and 10% for 2WikiMultiHopQA, respectively. As shown in Figure 2, rationale quality scores are higher for our method than the baseline. Since the task difficulty in 2WikiMultiHopQA is higher than that in HotpotQA, models tend to repeat sentences from the given context and fail to provide a final answer to the question when using the original CoT prompting. This phenomenon is not observed when applying our approach, leading to a substantial reduction in error rates. It also surpasses the performance of CoT prompting + SC and is on par with standard prompting in terms of EM and F1. For the answer inclusion score, LM-guided CoT prompting is slightly higher (1-2%) than standard prompting. Furthermore, LM-guided CoT prompting + SC achieves the highest performance across all settings.

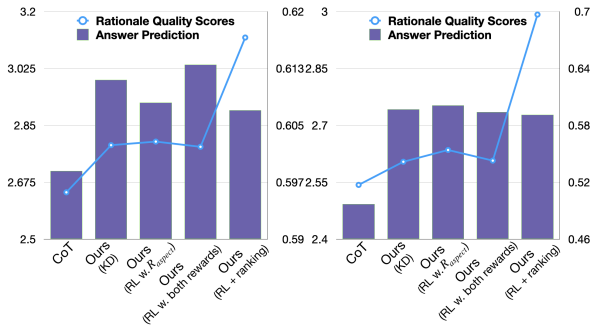


Figure 2: Average answer prediction performance (across three evaluation metrics) and average rationale quality scores (*i.e.*, R_{aspect}) for HotpotQA (left) and 2WikiMultiHopQA (right). The right y-axis represents the mean answer prediction scores and the left y-axis represents the mean rationale quality scores.

The implementation of RL empowers the model to achieve additional improvements in both rationale qualities and task performance. However, in line with Joshi et al. (2023)’s findings, a slight decrease

in task performance is observed at the cost of maximized rationale qualities when selecting top-quality rationales (Figure 2). There may be several underlying factors involved (*e.g.*, models’ unfaithfulness), but this is not the scope of this work.

5. Conclusion

LM-Guided CoT is a novel framework that decomposes a conventional CoT prompting into two steps using two models: (1) rationale generation and (2) answer prediction. This includes distilling the reasoning ability from a large LM to a small LM and further optimizing it with RL. The results reveal that our method outperforms all baselines, highlighting its potential to serve as an effective and resource-efficient approach to tackle challenges within the CoT prompting paradigm. Meanwhile, we also find that selecting top-quality rationales for answer prediction may not consistently boost task performance. This prompts the need to explore a more harmonious balance between LM-generated rationale utilities and overall task performance.

6. Limitations

While our framework is capable of accommodating diverse model combinations in a plug-and-play fashion, we have restricted our experimental reporting to FLAN-T5. In a similar vein, this work only explores the task of multi-hop QA, leaving an open question about generalizability to other reasoning tasks. We anticipate future investigations that will broaden the application of our approach across diverse domains that necessitate advanced reasoning. Lastly, due to resource constraints, we were unable to collect extensive human annotations for established aspect evaluation metrics.

7. Ethical Considerations

All the datasets that we use in our work are publicly available, and we have given appropriate credit to the original authors throughout the paper. We acknowledge that occasionally, the generated rationales may include non-factual and offensive statements. As we do not plan on distributing these artifacts, it effectively reduces the potential for harm.

8. Acknowledgements

We thank all the reviewers for providing valuable feedback.

9. Bibliographical References

- Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.
- Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.
- Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2022. Roscoe: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.
- Brihi Joshi, Ziyi Liu, Sahana Ramnath, Aaron Chan, Zhewei Tong, Shaoliang Nie, Qifan Wang, Yejin Choi, and Xiang Ren. 2023. Are machine rationales (not) useful to humans? measuring and improving human utility of free-text rationales. *arXiv preprint arXiv:2305.07095*.
- Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. *arXiv preprint arXiv:2205.11822*.
- Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.
- Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. *arXiv preprint arXiv:2306.14050*.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gptval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.
- Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.
- Yuhan Ma, Haiqi Jiang, and Chenyou Fan. 2023. Sci-cot: Leveraging large language models for enhanced knowledge distillation in small models for scientific qa. *arXiv preprint arXiv:2308.04679*.
- Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. 2023. Receval: Evaluating reasoning chains via correctness and informativeness. *arXiv preprint arXiv:2304.10703*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.
- 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, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022a. Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint arXiv:2212.10001*.
- Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Don-

ald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot prompting for textual reasoning. *Advances in neural information processing systems*, 35:30378–30392.

Ruo Chen 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*.

10. Language Resource References

Ho, Xanh and Nguyen, Anh-Khoa Duong and Sugawara, Saku and Aizawa, Akiko. 2020. *Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps*.

Yang, Zhilin and Qi, Peng and Zhang, Saizheng and Bengio, Yoshua and Cohen, William W and Salakhutdinov, Ruslan and Manning, Christopher D. 2018. *HotpotQA: A dataset for diverse, explainable multi-hop question answering*.

11. Appendices

11.1. Human Annotation for Rationale Quality Measurement

Annotation Details. Three researchers manually inspected 100 instances randomly sampled from training data. To ensure the quality and consistency of the annotation process, one researcher designed the annotation instruction describing the definition of each aspect (see Table 1) as well as 4 demonstration examples for each aspect category. Figure 3 displays demonstration examples for logicity. These examples are chosen randomly from the remaining training set, which was not included in the initial set of 100 annotation samples, and are manually annotated based on aspect descriptions. Upon the completion of annotation, we gauged the inter-rater agreement rates to validate the reliability of submitted annotation results by computing the average Fleiss’ Kappa coefficient (Fleiss, 1971). The mean Kappa score among the three annotators

across all aspect categories was 0.56, indicating a moderate agreement rate. Ultimately, we take the mode of annotated labels submitted by three annotators and consider it as a ground truth.

Analysis of Relationships between Aspect Types and Task Performance. Based on 100 labeled instances, we perform a post-hoc analysis to understand how the proposed aspects are related to answer prediction performance. The label distributions is as follows: correct (n=78) vs. incorrect (n=22). We compare the mean difference of evaluation scores based on correct & incorrect responses. As shown in Table 4, rationales associated with correct answer prediction are prone to have higher scores than those with incorrect prediction. In particular, coherence has the largest mean difference, followed by readability and logicity. For fluency and naturalness, the gap seems minimal. We further conduct statistical testing to verify their statistical significance. T-test results confirm that the mean difference observed in coherence ($p = 0.003$), readability ($p = 0.0002$), and logicity ($p = 0.05$) are statistically significant with $p < 0.05$.

11.2. Automatic Measurement for Rationale Quality

Method 1: Self-Evaluation from Large LMs. Fu et al. (2023) have demonstrated the emergent capabilities of large LMs in neural text evaluation, achieved through zero-shot instruction and in-context learning. The key idea is that, given the natural language description of desired task and evaluation aspects, large LMs such as GPT-3, OPT, FLAN-T5 can assess multi-dimensional text quality without any learning process. Motivated by this, we instruct FLAN-T5 XXL to evaluate 6 aspects of the machine-generated rationales. Similar to Fu et al. (2023)’s experiments, we investigate the performance of instruction-only (IST) prompting and instruction+demonstration (IDM) prompting. Let’s say q and r' denote a question and a machine-generated rationale that is yet to be evaluated. d represents the aspect definition of our interest from Table 1. A prompt template used for IST is:

Answer the question based on the provided information.

Question: Can the given reasoning d ? (a) Yes. (b) No.

Information:

Question: q
Reasoning: r'
Answer :

A prompt template for IDM is equivalent to IST, but with the inclusion of few task demonstrations. The

EXAMPLES

Logicity : measures whether the given reasoning is enough to reach the answer

Question : Who is the spouse of the composer of song You Keep Sending 'Em Over And We'll Keep Knocking 'Em Down?

Answer : Eileen Percy

Reasoning: Harry Rubenstein, known professionally as Harry Ruby (January 27, 1895 - February 23, 1974) was an American composer and screenwriter, who was inducted into the Songwriters Hall of Fame in 1970. He was married to silent film actress Eileen Percy.

Output : no

→ The output is 'no' because it is missing the information that Harry Rubenstein is the composer of song You Keep Sending 'Em Over And We'll Keep Knocking 'Em Down

Question : Which film has the director died first, Everybody Wins (Film) or Ringo And His Golden Pistol?

Answer : Ringo And His Golden Pistol

Reasoning : Ringo and His Golden Pistol is a 1966 Italian Spaghetti Western film directed by Sergio Corbucci and starring Mark Damon. Everybody Wins is a 1990 American mystery-thriller film directed by Karel Reisz, starring Debra Winger and Nick Nolte. The director of Ringo and His Golden Pistol died first, 1 December 1990. The director of Everybody Wins died last, 25 November 2002.

Output : yes

Figure 3: Demonstration example for "logicity" annotation.

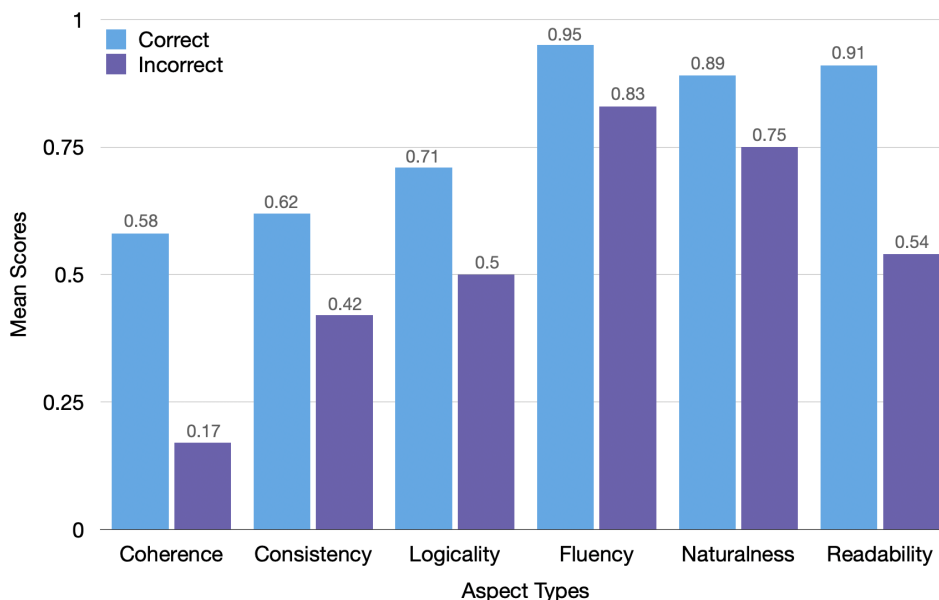


Figure 4: Mean scores of human annotation results by answer prediction correctness.

number of demonstrations range from 1 to 4. We use the same demonstrations from the annotation instruction (§11.1). Lastly, we evaluate IST in combination with self-consistency decoding, which involves sampling the prediction multiple times ($n = 10$) and taking the mode as a final predictions. To ensure that the rationale evaluation of FLAN-T5 XXL aligns closely with human annotation, we compute the macro F1 scores for 5 prompt-based experiments using 100 human-labeled examples (Table 4). In most cases except for fluency, IST prompting demonstrates the highest performance.

Method 2: Supervised Training with Human-

Annotated Data. Here we train a logistic regression classifier using 100 ground truth data from §11.1. This can be done by converting input data into TF-IDF vectors. Due to a small training data size, we resort to train two independent binary classifiers instead of having 6 models for each aspect type. While the first model is to predict if a given reasoning is logical, coherent, and consistent, the second one focuses on predicting whether the reasoning is fluent, natural, and readable. If at least one of the components is not satisfied, we consider it a negative label. The final label distribution is as follows: logicity & consistency & coherence (0:

Prompting		Coherence	Consistency	Logicity	Fluency	Naturalness	Readability
IST	IST	0.52	0.70	0.73	0.91	0.52	0.86
	IST + self-consistency	0.53	0.67	0.74	0.90	0.50	0.85
IDM	IDM (1 shot)	0.53	0.64	0.70	0.92	0.29	0.70
	IDM (2 shot)	0.53	0.62	0.73	0.91	0.24	0.69
	IDM (3 shot)	0.51	0.62	0.73	0.88	0.28	0.67
	IDM (4 shot)	0.52	0.62	0.70	0.89	0.36	0.50

Table 4: The macro F1 scores for 5 prompt-based experiments from Method 1, based on 100 human-labeled examples. Values in bold represent the best performance in each aspect.

Methods	Coherence & Consistency & Logicity				Fluency & Naturalness & Readability			
	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1
Method 1	0.62	0.62	0.6	0.6	0.7	0.7	0.81	0.67
Method 2	0.8	0.79	0.79	0.79	0.9	0.94	0.75	0.9

Table 5: Evaluation results (Method 1 vs. Method 2). Values in bold represent the best performance in each aspect.

60 vs. 1: 40), and fluency & naturalness & readability (0: 20 vs. 1: 80). We use 90% of dataset (n=90) for training and the remaining 10% (n=10) for evaluation.

Method 1 vs. Method 2. We attempt to assess which one is more suitable for providing rewards for RL. For a fair comparison, we use the same evaluation data that was used in Method 2. Since Method 1 has 6 aspect categories, we obtain individual results using the best performing approach and group them into two as we did for Method 1. Table 5 reports the accuracy, macro precision, macro recall, and macro F1 for Method 1 and Method 2. The results indicate that, although models from Method 2 are trained on a relatively small dataset, Method 2 is more aligned with human judgments compared to Method 1. Additionally, inference time using Method 2 is significantly faster than Method 1, making it easier to retrieve rewards in the RL setting. As a result, we use Method 2 for all our experiments.

11.3. Training Configuration

We provide a comprehensive description of the hyperparameters employed during model training.

Rationale Distillation. For rationale generation from the teacher model, we randomly sampled 15K examples from each of two datasets, resulting in a total of 30K examples. After filtering invalid rationales, the dataset was reduced to a total of 23K examples. 90% of data was used for training, and the remaining 10% was used for validation. For training, we used 8 NVIDIA Tesla V100 GPUs with 16GB configurations. Hyperparameters for training are as follows: 3e-3 for learning rates, 5 epochs, 64 batch size.

RL for Rationale Refinement. For RL, we uti-

lized the Huggingface’s TRL⁴ library that provides a set of tools to train transformer LMs with RL. Instead of using the same examples used for the rationale distillation, we selectively chose 5000 examples that FLAN-T5 XXL failed to answer the question correctly. The reason behind this choice was to increase the model’s exposure to challenging questions, thus increasing the likelihood of receiving more learning signals. We used 90% of data for training and the remaining 10% for validation. Training hyperparameters are as follows: 1.4e-5 for learning rates, 1 epoch, 16 batch size. For generation configurations, we set top_k as 0.0, top_p as 1.0, and enabled sampling.

⁴<https://huggingface.co/docs/trl/index>