Fine-tuning On Rationale Generation Improves Multi-Hop Reasoning in Small Language Models

Richard Mathews, **Adam Weinberger**, and **Meng-Kang Kao**

University of California, Berkeley

Abstract

In this paper we find self-ask and direct fine-tuning can elicit multi-hop reasoning in smaller language models. Both strategies show significant performance improvement over baselines. Qualitative analysis further reveals that the direct-tuned models, while improving model accuracy, lack the ability to perform true multi-hop reasoning compared with self-ask tuned models. On the other end, self-ask tuned models demonstrate their ability to follow the chain-of-thought (CoT) rationale to answer complex questions.

1 Introduction

Training language models to perform complex reasoning, allows them to answer multi-hop questions. For example, if we ask the question "Who was the president in office when the second Star Wars movie played in theaters?", the answer may not be easily found from the training data provided. However, with compositional reasoning, we can decompose a complex, multi-hop question into multiple, simpler sub questions whose answers can be composed to answer the overall question. In the previous question, we can break down the question to (1) When was the second Star Wars movie played in theaters? (The answer is 1980), and (2) Who was the US president in 1980 (the answer is Jimmy Carter). Therefore, the correct answer to the original question is Jimmy Carter. We refer to such decomposition as CoT rationale.

Our research is inspired by Measuring And Narrowing The Compositionality Gap in Language Models (Ofir, et al. 2022), which introduced a state-of-the-art CoT rationale structure called self-ask that elicited impressive few-shot multi-hop reasoning in larger GPT-3 models. We continue the research and evaluate the effectiveness of self-ask on various fine-tuned smaller language models.

First, the three base language models we selected for our experiment are T5, instruction-tuned (Flan-T5), and OPT model. Flan-T5 was first released in Scaling Instruction-Finetuned Language Models (Wei, Chung, et al. 2022), and is an enhanced version of T5 that has been fine-tuned to follow instructions. OPT, or Open Pre-trained Transformer Language Models, is an open-source language model that performs similarly to GPT-3.

Next, for each baseline language model, we further fine-tune two additional models with

Direct Tuning	Self-Ask Tuning
Prompt	Prompt
Facts: Fact #0: The film was written, adapted and directed by Russian-born Arcady Boytler. Fact #1: Boytler was born in Moscow, Russia. Question: Where was the director of film Heads Or Tails (1937 Film) born? Answer:	Fact #0: Mikko Esa Juhani Heikka(born 19 September 1944 in Ylitornio) is a Finnish former bishop of the Evangelic Lutheran Church. Fact #1: Scott Douglas Robbe is an American film, television, and theater producer/director. Question: Does Mikko Heikka have the same nationality as Scott Robbe? Are follow up questions needed here:
Training Label (Question Answering)	Training Label (Rationale Generation)
Moscow	Yes. Follow up: What is the country of citizenship of Mikko Heikka? Intermediate answer: Finnish Follow up: What is the country of citizenship of Scott Robbe? Intermediate answer: American So the final answer is: no

Table 1 Sample of direct and self-ask tuning prompting as well as the answers.

different datasets, one with direct prompting, and the other with self-ask prompting (see Table 1).

Lastly, we evaluate each language model with two different datasets, one with few-shot prompting, and one with zero-shot prompting. Quantitatively, we use F1 score and accuracy to measure performance among different models. We evaluate both the final answer as well as the generated rationale to evaluate the ability for each model to truly utilize self-ask mechanism. Additionally, from each fine-tuning category, we qualitatively analyzed top performing models to identify the differences.

In summary, our experiment shows that fine-tuning T5 models with fewer parameters on self-ask reasoning demonstrates impressive multi-hop reasoning capabilities.

2 The State of Reasoning in Language Models

(Ho, et al. 2020) showed that although many current models have defeated human performance on SQuAD, such performances do not indicate that these models can completely understand the text. Specifically, using an adversarial method, (Jia and Liang 2017) demonstrated that the current models do not precisely understand natural language.

(Ofir, et al. 2022) showed that with a self-ask rationale (Table 1), the compositionality gap reduced significantly with a large model (GPT3 Davinci, 175B parameter model), but with smaller models (Ada - 0.35B, or Babbage - 1.3B parameter models), the improvement was limited.

(Wei, Chung, et al. 2022) showed that with the instruction-tuned Flan-T5 models (as small as 80M parameters), the checkpoints have strong zero-shot, few-shot, and CoT abilities, and it out-performed prior public checkpoints such as T5.

(Fu, et al. 2023) fine-tuned a generic instruction-tuned model with arithmetic dataset and successfully trained Flan-T5 models to perform well on multi-hop question answering. The paper identified that in-context data preserves zero-shot ability, but not with zero-shot prompts.

The focus of our experiment is utilizing a state-of-the-art CoT method (self-ask) on three language models by inserting the desired thought pattern in the training labels and measuring if such fine-tuning will improve performance at multi-hop reasoning. Compared with other research, which focus on larger language models or postulate that small models struggle with compositional reasoning (Wei, Wang, et al. 2023), we are more interested in eliciting multi-hop reasoning of smaller language models.

3 Methodology

3.1 Data

Base Dataset: 2WikiMultiHopQA We use the 2WikiMultiHopQA (Ho, et al. 2020) dataset with some alterations. This dataset comprises approximately 200,000 multi-hop reasoning questions generated from Wikipedia, predominantly concerning the date of birth, lifespan, country of origin, and familial relationships of historical figures and celebrities. The dataset covers four categories of questions, including bridge-comparison, comparison, compositional, and inference. The test set does not have public answers, so we instead take the development set to be our test set.

Dataset Modifications We modify 2WikiMultiHopQA to create two datasets for fine-tuning. Note that we do these modifications programmatically in contrast to previous papers doing so manually (Mishra, et al. 2022) (Patel, et al. 2022) (Wei, Wang, et al. 2023). The two datasets are: direct no exemplars (Table 1 column 1) and self-ask with exemplars. The direct dataset "directly" provides the supporting facts and the question in the prompt. The self-ask with

Base Model Comparison				
Model	Architecture	Pre-training Task	Fine-tuning	Parameters
OPT-125m	Decoder only	Auto regressive next word prediction	None	125m
T5 small	Encoder – Decoder	Auto encoding denoising task	None	60m
Flan-T5 small	Encoder - Decoder	Auto encoding denoising task	Instruction Tuned	60m

Table 2 We are especially interested in decoder vs encoder-decoder, as well as the effect of instruction fine-tuning.

exemplars dataset provides two exemplars of self-ask reasoning before the supporting facts and question, and the prompt ends with the instruction "Are follow up questions needed here:". All data is lightly parsed to make it human-readable and appropriate for language models.

3.2 Models

Baseline Models We use three baseline models with different architectures and pre-training techniques to provide variation in our results and contrast with previous literature. Our baseline models are OPT, T5, and Flan-T5. See Table 2 above for detailed comparisons.

Fine-tuning Each baseline model is fine-tuned separately on the direct dataset and the self-ask with exemplars dataset. When fine-tuning on the latter dataset, we use the full self-ask rationale as the target. We hypothesize this will better train the model to perform multi-hop reasoning. This contrasts with other compositional reasoning papers which either focus on prompt engineering or do not use the entire rationale as the target. We use all the default parameters except that we train for two epochs and use batch size 32. We end up with three models for each of the three model families: baseline (not tuned), direct, and self-ask. This gives us nine models in total.

3.3 Evaluation

We use F1-1 (1-gram) and F1-2 (2-gram) to evaluate rationale generation, and accuracy to evaluate question answering. We provide secondary metrics in the appendix.

We use F1-1 and F1-2 because most of our self-ask rationale targets are a few dozen tokens long so we want to balance penalizing a model for extraneous text while simultaneously rewarding a model for repeating the correct answer.

We evaluate a model as correct if the true answer appears anywhere in the generated answer. We then calculate accuracy in the normal way. We choose accuracy as an important metric because it is the most straightforward way to assess whether the model answers the question correctly. However, a response containing the correct answer may be quite wrong, which is why we have the F1 measures and manually inspect the answers. We discuss this further in our results section.

Each model is evaluated on two datasets with and without exemplars (i.e. few-shot and zero-shot) giving us 18 total evaluations.

4 Results

We report results on both the question answering task and the rationale generation task for the T5 and Flan-T5 families.[1]

4.1 Multi-hop Reasoning

Baseline Comparisons We contextualize our fine-tuning results against our baseline set of T5 models. In zero-shot setting, the direct-tuned versions of T5 and Flan-T5 show significant improvements compared to their baseline counterparts (+38% and +14.5%, respectively), outperforming all baseline models.[2] As for the self-ask-tuned models, they outperform their baseline counterparts in both zero-shot and few-shot regimes. Most notably, the self-ask Flan model realizes a **24.3%** increase in accuracy over the baseline Flan model.

Fine-tuning Comparisons The best performing model in both zero-shot and few-shot is the self-ask-tuned Flan-T5-small model, scoring 7% higher accuracy over the best direct-tuned model.

We also examine the breakdown of accuracies by question type for the three fine-tuned variants of T5-small (Figure 1) and the two fine-tuned variants of Flan-T5-small (Appendix A).[3]

We find T5-small has impressive performance on compositional questions off-the-shelf and the instruct-tuned version (Flan-T5) raises accuracies in comparison questions. When directly fine-tuned on multi-hop question-answer pairs, T5 shows even greater improvements in comparison and a remarkable jump in inference (+68% vs T5, +59% vs Flan-T5). The self-ask fine-tuned T5 achieves even greater improvements across all questions over the direct-tuned version, but it appears most of the performance gains come from fine-tuning on a multi-hop reasoning QA dataset and not fine-tuning on rationale generation.

[1] The OPT family, in all cases, scores 0% accuracy since they only repeat the prompt and don't attempt to answer the question or generate a rationale, so those results are not reported.

[2] Direct-tuned T5 and Flan T5 perform worse than baseline counterparts in the few-shot setting, indicating the presence of examplars is performance-denigrating.

[3] For fair comparison, we use the best performer out of zero-shot vs few-shot settings for each category.

	Accuracy (QA)	F1-1 (Rationale)	F1-2 (Rationale)
T5-Small (Zero-Shot)	32.9	-	-
T5-Small (Few-Shot)	24.8	0.05	0.02
Flan T5-Small (Zero-Shot)	51.7	-	-
Flan T5-Small (Few-Shot)	53.6	0.08	0.05
Direct-tuned T5-Small (Zero-Shot)	70.9	-	-
Direct-tuned T5-Small (Few-Shot)	10.6	0.02	0.01
Direct-tuned Flan T5-Small (Zero-Shot)	66.2	-	-
Direct-tuned Flan T5-Small (Few-Shot)	47.0	0.08	0.05
Self-Ask T5-Small (Zero-Shot)	40.9	0.66	0.56
Self-Ask T5-Small (Few-Shot)	74.8	0.96	0.94
Self-Ask Flan T5-Small (Zero-Shot)	**73.2**	**0.94**	**0.91**
Self-Ask Flan T5-Small (Few-Shot)	**77.9**	**0.97**	**0.95**

Table 3: (a) Multi-hop reasoning results on modified 2WikiMultiHopQA (REF), reported values are accuracy at answering the question given relevant context only; (b) Rationale generation results for models that are either fine-tuned on the self-ask rationale or provided in-context examples of self-ask rationale, reported values are unigram F1 score and bigram F1 score. Best performances for zero/few-shot are bold.

In the fine-tuning results for the Flan-T5 model, we find a different outcome. Fine-tuning Flan-T5 directly on question-answer pairs only improves performance on composition and inference questions, but fine-tuning Flan-T5 on rationale generation leads to significant performance gains on bridge comparison and comparison questions (+17% and +29%, respectively), in addition to the performance gains achieved by direct-tuned Flan-T5.

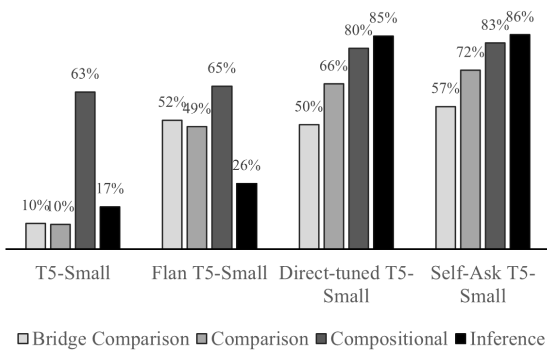

Figure 1: Accuracy results on the four types of multi-hop questions for baseline T5-small and the three fine-tuned versions of T5-small.

4.2 Rationale Generation

In the rationale generation tasks, the objective is to elicit self-ask reasoning in models with the hope of improving multi-hop reasoning. We report F1 scores comparing generated text to target text (self-ask rationale) in Table 3.

All models that are not fine-tuned on rationale generation exhibit no capabilities in providing self-ask reasoning when provided two in-context examplars of self-ask.

We find that self-ask-tuned models exhibit a perfect 100% rate at responding to questions using self-ask reasoning in the few-shot regime, but a surprising finding is that they also exhibited perfect (Flan T5) or near perfect (T5) self-ask response rates in the zero-shot regime.

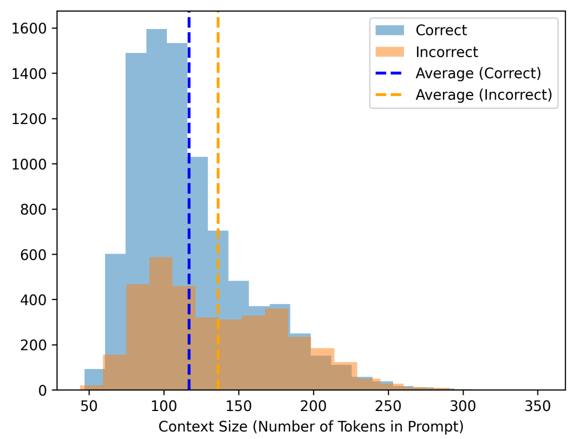

Figure 2: Distributions of the number of prompt tokens for questions that the zero-shot direct-tuned T5-small model answers (i) correctly (blue); (ii) incorrectly (orange).

5 Analysis

Armed with empirical evidence of the effectiveness of fine-tuning small encoder-decoder language models directly on multi-hop question-answer pairs and on self-ask rationale generation, we set out to understand *how* they are different and *what* their weaknesses are. We present our findings from quantitative and qualitative analyses on the two best models from each fine-tuning category: zero-shot direct-tuned T5-small and few-shot self-ask Flan T5-small.

5.1 Correlations in Outcomes

Our results in Figure 1 show the best direct-tuned model exhibits a very similar performance profile to the best self-ask model on the macro-level. However, at the micro-level, the models perform quite differently.

The correlation of their question-level outcomes (right vs wrong) is only 0.38, revealing that they don't get the same questions right nor the same questions wrong. In fact, the direct-tuned model correctly answers **38%** of the self-ask misses, and the self-ask model correctly answers **53%** of the direct-tuned misses. It is apparent that these two fine-tuning strategies lead to different strengths and weaknesses, which is what we explore in follow-up sections.

5.2 Weaknesses in Direct-Tuned Models

Wrong Answers A qualitative inspection of the challenging questions for the direct-tuned T5 model reveals the model simply provides the wrong answer, suggesting the source of its errors is a failure to engage multi-hop reasoning.

Large Context A follow-up quantitative analysis (Figure 2) reveals the model is particularly sensitive to larger context sizes (3+ facts), which indicates it disproportionately struggles on questions with more than two "hops". We do not observe this weakness in the self-ask model.

5.3 Weaknesses in Self-Ask Models

Self-Questioning Loops In our qualitative analysis of the questions that the self-ask model struggles on, we observe a unique weakness, which we call "self-questioning loops." Rather than providing the wrong answer, the model enters an unending loop when trying to answer one of its intermediate questions, as shown in Figure 3.

The pattern amongst the questions that trigger these loops is the presence of some corruption in the prompt itself, such as a typo or mislabeling. With some minor ablations, we confirm that fixing the fact corruption eradicates the self-questioning loop and, in many cases, results in a correct answer. We also confirm that the model only enters a self-questioning loop *when it tries to answer an intermediate question that calls upon corrupted information*. When inserting the same type of corruption in information that is irrelevant to the question, the model is *unaffected*.[4]

This is a major finding because it indicates the self-ask model is attempting to solve the problem by breaking it down into simpler questions, and if

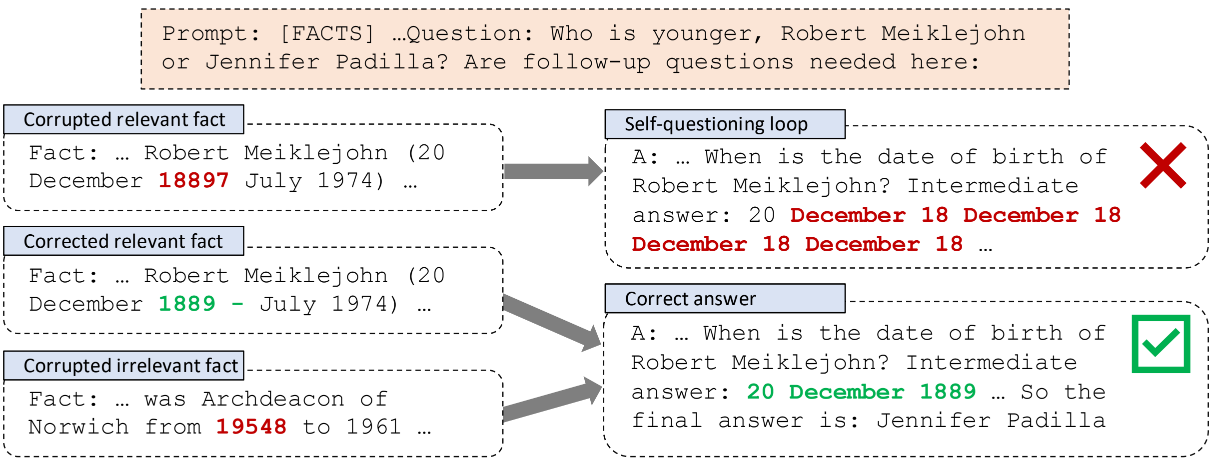

Figure 3: Qualitative example of the effect of fact corruption on the self-ask model. When facts that are relevant to the question have typos or errors (upper left), the self-ask model enters a "self-questioning loop" when accessing the corrupted information. Fixing the error (middle left) leads to a correct response, and inserting the same error in irrelevant information (bottom left) does not induce a self-questioning loop.

[4] We also confirm the same patterns in the zero-shot setting.

the information needed to answer the simpler question is corrupted, the model "breaks".[5]

On the other hand, we find the direct-tuned model *consistently answers these corrupted questions correctly*, indicating it is agnostic to errors in critical pieces of information and is figuring out an alternative path to answering the question besides multi-hop reasoning.

6 Discussion

It is obvious fine-tuning models on multi-hop reasoning QA datasets significantly improves multi-hop reasoning, but the interesting question pertains to the advantage of fine-tuning a model to generate a rationale (text generation task) over fine-tuning a model to directly answer a question (QA task). Given the decoder-only models we test fail to learn the task, we focus on the encoder-decoder architecture.

We postulate a self-ask encoder-decoder model has the upper hand because it has a larger "attentional capacity", i.e. it can attend to its intermediate answers in decoder self-attention when generating the final answer (evident in Figure 9) in addition to the encoded prompt, whereas a model trained to directly answer the question is forced to attend over the entire input via cross-attention. There are two key pieces of evidence to support this idea.

1. The direct-tuned model struggles more on larger context sizes in the encoder compared to the self-ask model. (App. B)

2. Inspection of the attention weights reveals that both models have similar cross-attention patterns, but very different decoder self-attention patterns.

We also find it plausible that the self-ask models develop a true multi-hop reasoning capability, whereas the direct-tuned models learn alternative patterns unrelated to fact composition given its limited attentional capacity. Our key piece of evidence supporting this argument is the contrast in their response to questions with corrupted relevant facts. Self-ask models enter self-questioning loops when trying to make sense of information required to compose the facts, whereas the direct-tuned models appear agnostic to these fact corruptions, suggesting they arrive at answers by means other than composing facts. It's entirely possible the models trained on question-answer pairs learn structural patterns given our dataset is generated algorithmically, and their weaknesses are explained by disruptions to these structural patterns (such as the inclusion of more than two facts).

In summary, the phenomenon of self-questioning loops in the model trained on rationale generation suggests there is some degree of reasoning via fact composition. While the two models might appear to have similar performances on paper, the actual multi-hop reasoning capabilities may be quite different.

Conclusion

In this paper, we investigate how a fine-tuning approach can elicit multi-hop reasoning in smaller language models (<1B parameters). We compare two fine-tuning strategies: direct-tuning (directly fine-tune on question-answer pairs) and self-ask-tuning (fine-tune on question-rationale pairs, i.e. rationale generation). Our findings demonstrate that both strategies greatly improve over baselines for encoder-decoder transformers. We also explore why models fine-tuned to generate self-ask rationales might be better multi-hop reasoners than models trained to directly answer the question, drawing on insights from qualitative and quantitative analyses.[6]

References

Fu, Yao, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. *Specializing Smaller Language Models towards Multi-Step Reasoning*.

Ho, Xanh, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. "Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps." *28th International Conference on Computational Linguistics*. Barcelona. 6609–6625.

Jia, Robin, and Percy Liang. 2017. *Adversarial Examples for Evaluating Reading Comprehension Systems.*

Mishra, Swaroop, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. "Reframing Instructional Prompts to GPTk's

[5] Appendix E provides a visual of the attention patterns before and after fixing the error.

6 We discuss potential opportunities for follow-up research in Appendix.

Language." Dublin: Association for Computational Linguistics. 589–612.

Ofir, Press, Muru Zhang, Sewon Min, and Ludwig Schmidt. 2022. *MEASURING AND NARROWING THE COMPOSITIONALITY GAP IN LANGUAGE MODELS*.

Patel, Pruthvi, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. *Is a question decomposition unit all we need?*

Wei, Jason, Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, et al. 2022. *Scaling Instruction-Finetuned Language Models*.

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

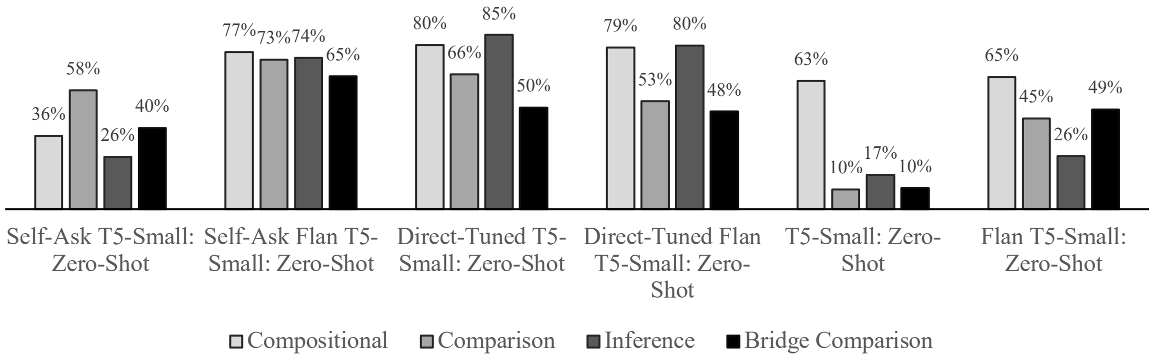

Figure 4: Performance breakdown by question type for models in zero-shot regime.

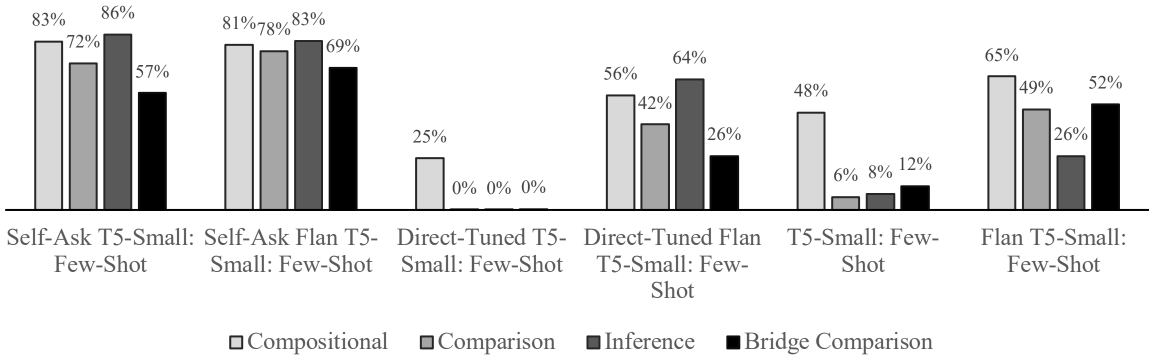

Figure 5: Performance breakdown by question type for models in the few-shot regime.

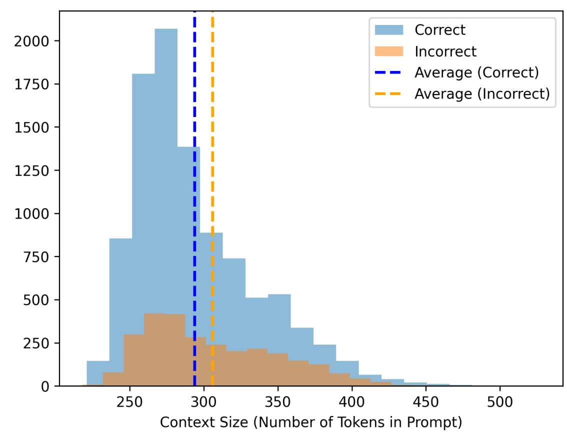

Figure 6: Distributions of the number of prompt tokens for questions answered by the few-shot self-ask-tuned Flan T5-small model (i) correctly (blue); (ii) incorrectly (orange).

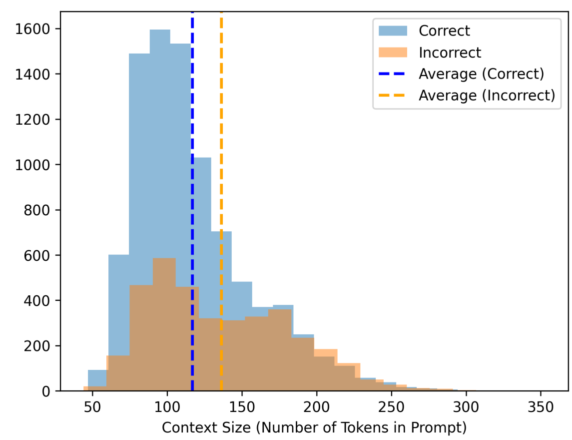

Figure 7: Distributions of the number of prompt tokens for questions answered by the zero-shot direct-tuned T5-small model (i) correctly (blue); (ii) incorrectly (orange).

A Performance Breakdown by Question

Figure 4 and Figure 5 show the accuracies on the multi-hop reasoning QA dataset by question category for models in the zero-shot and few-shot regimes, respectively.

B Effect of Context Size

Figure 6 and Figure 7 show that the best self-ask-tuned model is less affected by context size than the best direct-tuned model, indicating fine-tuning on rationale generation leads to improved performance on more complex reasoning questions. The prompts with larger context sizes contain more than 2 facts in the context.

C Zero-Shot vs Few-Shot Comparisons

The presence of in-context self-ask examplars yields mixed results. The models fine-tuned directly on question-answering exhibit substantial deterioration when provided demonstrations, dropping as much as 60% in the case of direct-tuned T5. The self-ask models exhibit the opposite

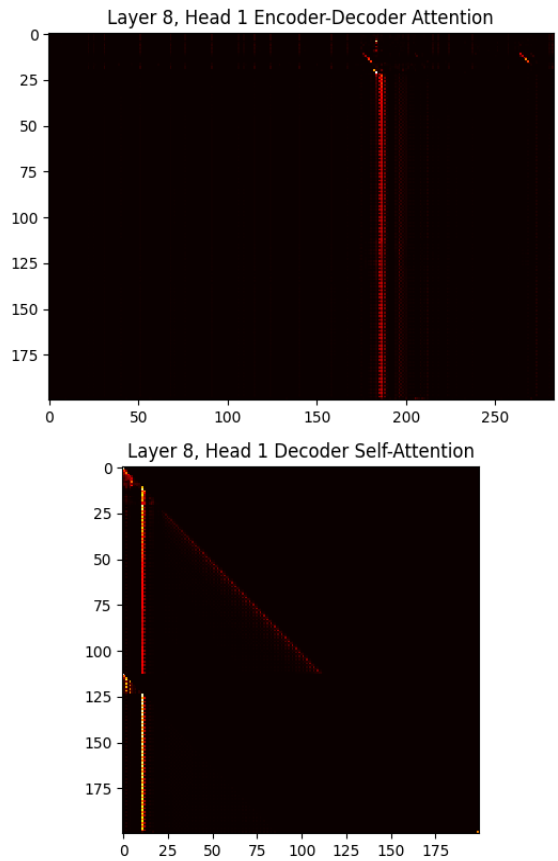

Figure 8: Attention patterns when the self-ask model enters a self-questioning loop show it attends to the same tokens in a loop. The y-axis is the generated token index, and the x-axis is the attention index (y attends to x).

pattern, where examplars significantly improve performance of self-ask T5 (+34%) but only moderately improve performance of self-ask Flan T5 (+4.7%).

D Sensitivity Analysis

In our qualitative ablation study, we discover some additional weaknesses shared by both fine-tuned models (direct-tuned and self-ask-tuned) that are not central to the main theme of the paper but may still be interesting to readers.

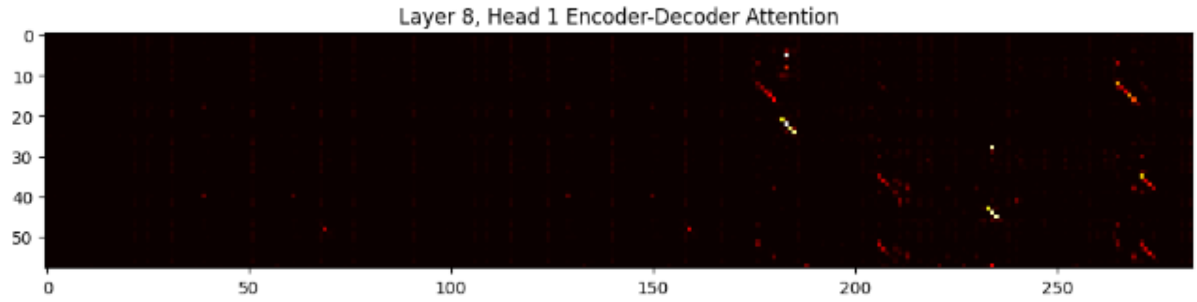

Figure 10: The cross-attention pattern of a self-ask model after correcting a typo in a relevant fact shows it returns to normal behavior (attending to facts and the question as it generates a response). The y-axis is the generated token index, and the x-axis is the attention index (y attends to x).

Sensitivity to Order A study on the effect of rearranging the order of facts given to the models shows both are not robust to the arrangement of contextual facts. Swapping facts consistently leads to the model changing its answer.

Sensitivity to Irrelevant Facts We investigate the response of the self-ask model to the insertion of irrelevant facts in its prompt, and find it always leads to nonsensical reasoning. This is not surprising considering we design the training to teach the model to assemble every fact it is provided, not to additionally identify which facts are relevant. We also observe the direct-tuned model changes its answer when irrelevant facts are inserted in the context.

E Attention Patterns in Self-Ask Models

To better understand the inner workings of the self-ask-tuned encoder-decoder transformer models, we visualize the attention patterns in the encoder-decoder and decoder.

Figure 10 shows cross-attention in the self-ask model when it answers a question correctly, Figure 8 shows cross-attention and decoder self-attention when the model enters a self-questioning loop, and Figure 9 shows decoder self-attention when the model answers a question correctly. Noteworthy observations we make are:

- Most of the model's attention is on specific parts of the facts and question, and not so much on the self-ask examplars.
- The model appears to attend to its intermediate reasoning steps as it generates the output.

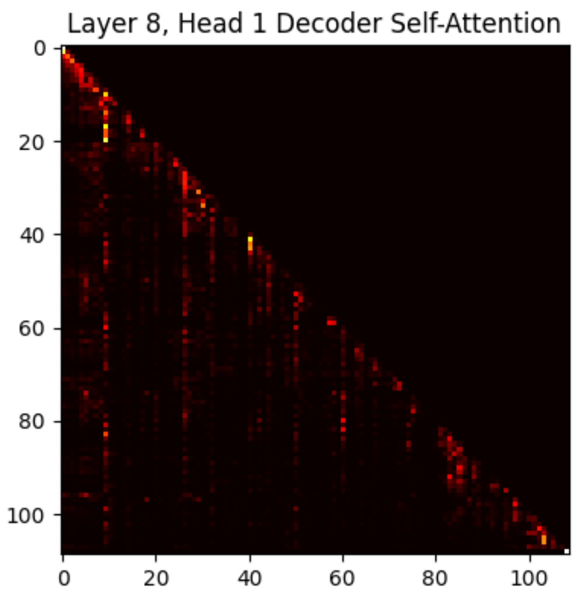

Figure 9: Decoder self-attention pattern of a self-ask model when it answers a question correctly. The y-axis is the generated token index, and the x-axis is the attention index (y attends to x).

- In a self-questioning loop, the model is repeatedly attending to the same tokens over and over until it hits the maximum generation length.

F 2WikiMultiHopQA

2WikiMultiHopQA has 167,454 training samples, and 12,576 development and test samples each. Each sample has the following keys:

- _id: a unique id for each sample
- question: a string
- answer: an answer to the question. The test data does not have this information.

- supporting_facts: a list, each element is a list that contains: [title, sent_id], title is the title of the paragraph, sent_id is the sentence index (start from 0) of the sentence that the model uses. The test data does not have this information.

- context: a list, each element is a list that contains [title, setences], sentences is a list of sentences.

- evidences: a list, each element is a triple that contains [subject entity, relation, object entity]. The test data does not have this information.

- type: a string, there are four types of questions in our dataset: comparison, inference, compositional, and bridge-comparison.

- entity_ids: a string that contains the two Wikidata ids (four for bridge_comparison question) of the gold paragraphs, e.g., 'Q7320430_Q51759'.

- until it hits the maximum generation length.

G Limitations

Given our findings in the analysis, it is quite possible we underestimate the performance of self-ask-tuned models given a nontrivial portion of its answers marked "wrong" are due to corruption issues in the data construction process and not to deficiencies in the model. We believe this limitation only applies to self-ask models as the baseline models and direct-tuned models do not exhibit this sensitivity to fact corruption.

H Future Work

We believe there is plenty of follow-up work necessary to understand the true capabilities of models fine-tuned on rationale generation compared to models fine-tuned directly on question-answering. Specific areas would be to study generalization to unseen tasks, the effect of model size, and the effect of fine-tuning to not only compose facts, but to also identify which ones are relevant to the question.

Type of Text	Font Size	Style
paper title	15 pt	**bold**
author names	12 pt	**bold**
author affiliation	12 pt	
the word "Abstract"	12 pt	**bold**
section titles	12 pt	**bold**
document text	11 pt	
captions	10 pt	
abstract text	10 pt	
bibliography	10 pt	
footnotes	9 pt	

Table 4: Font guide.