Exploring the Challenges of Multi-Step Logical Reasoning with Language Models: A Few-Shot Approach to Explainable Entailment Trees

Bowen Gao[1] Shotaro Kitamura[1] Naoya Inoue[1,2]
[1]JAIST [2]RIKEN
{gaobowen,s2210055,naoya-i}@jaist.ac.jp

Abstract

An Entailment Tree is a type of explainable entailment task that requires **L**arge **L**anguage **M**odels (LLMs) to produce intermediate hypotheses and reasoning steps. Previous studies have focused on using fine-tuned models, including the T5 model, for this purpose. However, fine-tuning typically involves high computational training costs and necessitates a comprehensive dataset to be effective. This paper explores the use of a decoder-only model combined with few-shot learning. This technique is, facilitated by a prompt, and is implemented to generate the intermediate hypothesis. Our experiments show that it is challenging to achieve results superior to those obtained using fine-tuned models, including the T5 model. We conducted an analysis to understand why decoder-only models do not excel in this area, and we hope that our findings can aid other researchers in investigating the potential of decoder-only models for explainable entailment tasks.

1 Introduction

The Entailment Tree [1] is an advanced explainable entailment task, incorporating multi-premise steps from facts to hypothesis, offering more meticulous and rigorous reasoning than prior tasks like reading comprehension [2, 3], explainable fact verification [4, 5] or open-domain QA [6, 7]. Its main challenge lies in methodically tracing reasoning from facts to conclusion, rather than just presenting textual evidence.

Initially, researchers utilized the All-At-Once generation method [8], based on the fine-tuned T5 model [9]. However, this approach often failed to generate valid intermediate hypotheses at each step, leading to ineffective Entailment Tree construction. To address this, a step-by-step method was considered, incorporating verification mechanisms like the Verifier [10], Retrieval-generation Reasoner [11] or Reinforcement Learning [12] to enhance the validity and overall performance of each step in the Entailment Tree.

Additionally, extensive research has focused on fine-tuning these models with smaller datasets and applying transfer learning to new problems. Employing few-shot learning enhances performance on specific tasks with limited data. Hence, we plan to use GPT-3.5 for in-context learning in Entailment Tree tasks, anticipating that GPT-3.5 will be effective for this purpose.

Inspired by the Chain-of-Thought [13] approach, we believe that instruction fine-tuned language models are well-suited for multi-step reasoning tasks. Therefore, we aim to employ these fine-tuned models for the Entailment Tree task, which involves more reasoning steps and a more structured approach.

The experiment was conducted using GPT-3.5 to construct the Entailment Tree. After evaluating with the Tree Alignment Algorithm, the results, although not outperforming the fine-tuned model, were notable. Our method, utilizing only few-shot learning, remains comparable to the fine-tuned model in terms of computational and training cost efficiently.

Our main contributions are below:

- We utilize the use of decoder-only model with few-shot learning for constructing the Entailment Tree.
- We provide analysis and exploit challenges of multi-step logical reasoning with Language Models.

We hope that our research can offer a feasibility to use few-shot in-context learning to deal with explainable entailment task. Besides, we hope that our insights could help other researchers explore the use of few-shot in-context learning in dealing with the explainable entailment task.

2 Methodology

2.1 All-At-Once

We employed an instruction fine-tuned language model to generate the Entailment Tree using our specific prompt. Our objective was for this fine-tuned model to generate the Entailment Tree in an All-At-Once manner. The primary content of our prompt is as follows:

```
'''
Given a hypothesis, generate a proof tree from given context.
Examples
hypothesis: the difference between atomic mass and atomic number is the number of the neutrons in an element
context: sent1: atomic mass is determined by the number of protons and neutrons sent2: atomic number is only determined by the number of proton of an element
proof: sent1 & sent2 -> hypothesis;
'''
```

2.2 Step-by-step

Due to the meticulous nature of the steps in the Entailment Tree, we hypothesized that all Entailment Trees could be constructed as binary trees. This means that each step could be generated from a pair of supporting facts using GPT-3.5 (Figure 1).

Besides, inspired by previous work on generating the Entailment Tree step-by-step [9, 10, 11, 14], we aimed to establish rules to verify the validity of generated intermediate steps. Additionally, we planned to utilize the beam search algorithm to iteratively generate the Entailment Tree step-by-step.

As illustrated in Figure 2, our first step was to combine and arrange all possible pairs of facts as candidates. Then, using a prompt, we instructed GPT-3.5 to deduce an intermediate hypothesis for each pair of facts. To ensure that each step brought the generated intermediate hypothesis closer to the final hypothesis, we intended to use Sentence-Bert to calculate the similarity score between the intermediate and final hypotheses. We planned to extract the top-K intermediate facts with the highest similarity scores. Lastly, to ensure that the final Entailment Tree was the optimal solution, we employed the beam search algorithm for iteratively generation.

We are also show the part of our prompt to make GPT-3.5 generate the intermediate hypothesis for chosen facts:

```
'''
Tasks
We are trying to prove a hypothesis from facts step-by-step.
Create an intermediate hypothesis based on supporting facts.
Rules
1. The created intermediate hypothesis should be as short as possible.
2. The created intermediate hypothesis must be based on inferences.
'''
```

We want to make GPT-3.5 conclude the intermediate hypothesis based on inferences.

3 Evaluation

We evaluated our methods and used the evaluation results to compare with fine-tuned T5 output as shown in Table 1. The gold tree is obtained from dataset, EntailmentBank [1].

3.1 Setup

In Entailment Tree task, it consists of average 7.6 nodes (supporting facts) across 3.2 entailment steps (multi-hop tasks are usually only consist of 2-3 supporting facts). Besides, Entailment Tree's reasoning steps are more meticulous which means in each supporting fact, it is always a simple short sentence, and this sentence is typical in proof generation. It is mainly focusing on two dimensions. The one is about the quality of the created intermediate hypothesis and another is about the final Entailment Tree structure.

As the result, we would like to evaluate these two dimensions (Intermediates and Steps) with F1 score and All Correct score. It is same with the evaluation metric which is always applied on the Entailment Tree task [1].

3.2 Results

Before evaluating our methods, we used GPT-3.5 to evaluate the single step's reasoning ability to verify its capacity

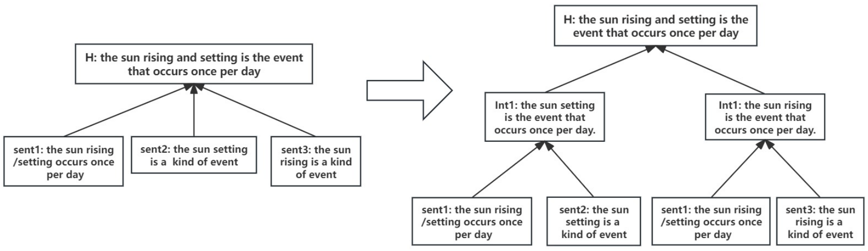
H: the sun rising and setting is the event that occurs once per day
sent1: the sun rising /setting occurs once per day
sent2: the sun setting is a kind of event
sent3: the sun rising is a kind of event
H: the sun rising and setting is the event that occurs once per day
Int1: the sun setting is the event that occurs once per day.
Int1: the sun rising is the event that occurs once per day.
sent1: the sun rising /setting occurs once per day
sent2: the sun setting is a kind of event
sent1: the sun rising /setting occurs once per day
sent3: the sun rising is a kind of event

Figure 1 We assumed that all the Entailment Trees' step can be explained by the binary tree

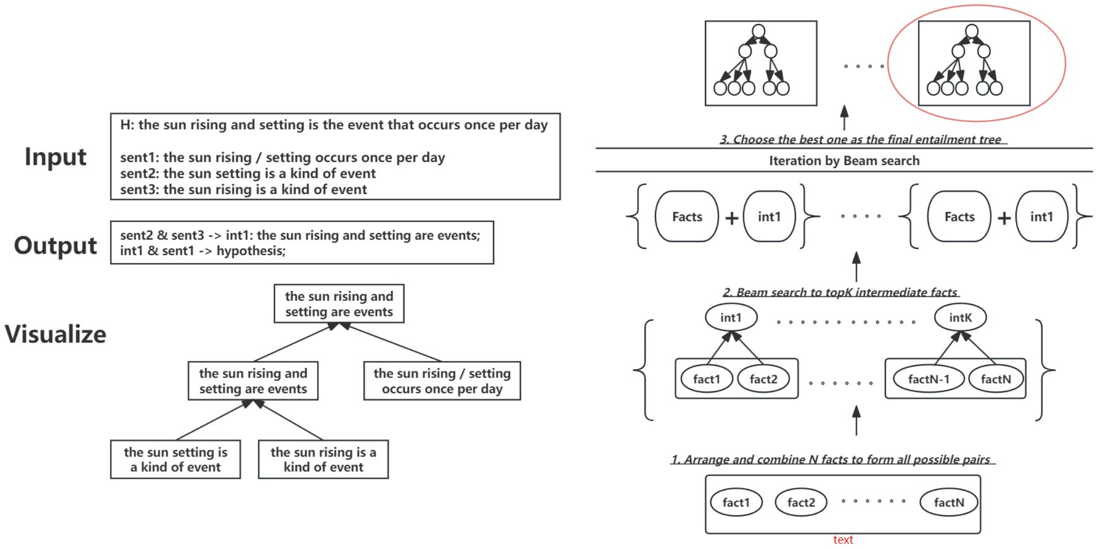
Input
H: the sun rising and setting is the event that occurs once per day
sent1: the sun rising / setting occurs once per day
sent2: the sun setting is a kind of event
sent3: the sun rising is a kind of event
Output
sent2 & sent3 -> int1: the sun rising and setting are events;
int1 & sent1 -> hypothesis;
Visualize
the sun rising and setting are events
the sun rising and setting are events
the sun rising / setting occurs once per day
the sun setting is a kind of event
the sun rising is a kind of event
3. Choose the best one as the final entailment tree
Iteration by Beam search
Facts
int1
Facts
int1
2. Beam search to topK intermediate facts
int1
intK
fact1
fact2
factN-1
factN
1. Arrange and combine N facts to form all possible pairs
fact1
fact2
factN
text

Figure 2 The structure of step-by-step generating the Entailment Tree with GPT-3.5

to handle explainable entailment task. We extracted all single steps from the gold tree and used GPT-3.5 to generate intermediate hypotheses based on given facts for each step. We assumed that in each step, the facts are always same with the gold one, we only used GPT-3.5 to generate intermediate hypotheses. According to the previous work's evaluation metric, GPT-3.5's score is 0.879 for intermediate hypothesis generation, indicating its capacity to manage the Entailment Tree task.

We compared our method which used the GPT-3.5 model with All-At-Once generation method using the fine-tuned T5 model. As shown in Table 1, neither the All-At-Once nor the Step-by-Step method with GPT-3.5 could outperform the fine-tuned T5 model.

We then utilized our methods in Python to generate the Entailment Tree with the GPT-3.5 model. Additionally, we used Graphviz library to visualize the Entailment Tree (Figure 3) for ease of analysis of its structure and content. Regard Figure 3, we observed that step1,2 and 3 all used conjunctions to infer new intermediate hypotheses, ultimately generating a well-structured Entailment Tree.

Table 1 Evaluation result. The evaluation result mainly focus on leaves, steps, intermediates and overall proofs. (L=Leaves , S=Steps, I=Intermediates, O=Overall Proofs.

	L	S	I	O
Methods	F1	F1	F1	AllCorrect
Fine-tuned T5	0.99	0.52	0.71	0.35
All-At-Once (GPT-3.5)	0.96	0.36	0.65	0.27
Step-by-step (GPT-3.5)	0.96	0.38	0.59	0.28

4 Analysis

From Experiment and its results, we concluded that the GPT-3.5 model has certain limitations in dealing with explainable entailment tasks like the Entailment Tree, and thus it cannot surpass the performance of the fine-tuned T5 model.

To understand why GPT-3.5 could not outperform the fine-tuned T5 model, we manually evaluated the quality of conclusions generated by the GPT-3.5 model and identified errors with specific reasons. These analyses led us to several conclusions.

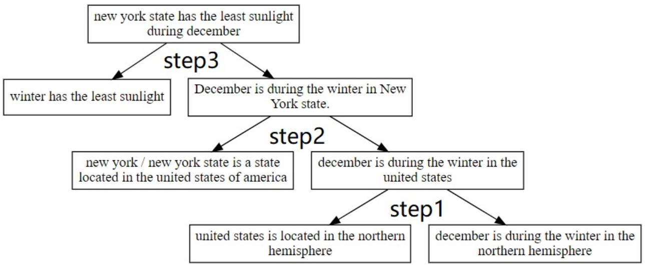
new york state has the least sunlight during december
step3
winter has the least sunlight
December is during the winter in New York state.
step2
new york / new york state is a state located in the united states of america
december is during the winter in the united states
step1
united states is located in the northern hemisphere
december is during the winter in the northern hemisphere

Figure 3 The visualized Entailment Tree

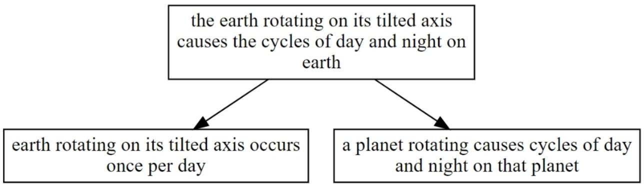
the earth rotating on its tilted axis causes the cycles of day and night on earth
earth rotating on its tilted axis occurs once per day
a planet rotating causes cycles of day and night on that planet

Figure 4 Example of the common sense problem

Common sense Based on the results of Experiment 2, we observed that approximately 27% of the samples showed GPT-3.5 not performing logical reasoning solely based on the given premises/facts. Instead, It used its own knowledge to draw conclusions.

For instance, as shown in Figure 4, the model could conclude without an important fact - ”earth is a kind of planet.” - derived from supporting facts. It was challenging to make GPT-3.5 understand the importance of this fact for the Entailment Tree’s generation, mainly because the model did not consider it relevant.

Invalid reasoning Based on the results from Experiment 2, among the 40 samples tested, we found that approximately 27% of the samples demonstrated that GPT-3.5 does not perform valid reasoning. As evidenced in Figure 5, the intermediate hypothesis generated by GPT-3.5 was found to be unreasonable.

Useless conclusion Based on the results of Experiment 2, among 40 samples, we found that approximately 20% of the samples indicate that the intermediate hypotheses generated by GPT-3.5 are ineffective for further generation in the Entailment Tree task. Specially, these hypotheses are unhelpful because they merely replicate one of the given facts. For example, when provided with the facts [’mercury is a kind of planet’] and [’mercury is located in the solar system’], GPT-3.5 generated the conclusion [’mercury is located in the solar system’], which is essentially just a repetition of the fact [’mercury is a kind of planet’].

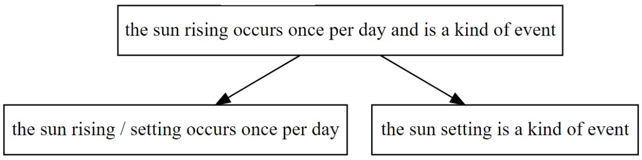
the sun rising occurs once per day and is a kind of event
the sun rising / setting occurs once per day
the sun setting is a kind of event

Figure 5 Example of the invalid reasoning

5 Conclusion and future work

Contribution In our research, we exclusively used instruction fine-tuned language models for building the Entailment Tree. While our method didn’t outperform fine-tuned models, it offers comparable results to the T5 model in terms of computational efficiency. We’ve analyzed and identified reasons for its limitations and plan enhancements in future work. Recognizing the complexity of the Entailment Tree task, which demands careful step-by-step reasoning.

Future work To rely solely on instruction fine-tuned models for generating the Entailment Tree and potentially outperforming fine-tuned models like T5, we need to address specific challenges:

- Establish a strict and clear mechanism that enables the model to discern which pairs of facts can produce a valid intermediate hypothesis, avoiding invalid reasoning or mere replication of facts.
- Implement a rule that helps the model recognize instances where relying solely on common sense leads to errors, guiding it towards more accurate and logical reasoning.

we believe that the instruction fine-tuned language model will show improved performance in handling complex reasoning problems, such as those involved in the Entailment Tree task.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 19K20332.

We thank Surawat Pothong, Yufeng Zhao and Irfan Robbani for their valuable comments.

References

[1] Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. Explaining answers with entailment trees. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott

Wen-tau Yih, editors, **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 7358–7370, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[2] Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. Multi-hop reading comprehension through question decomposition and rescoring. **arXiv preprint arXiv:1906.02916**, 2019.

[3] Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. **arXiv preprint arXiv:1905.07374**, 2019.

[4] Huanhuan Ma, Weizhi Xu, Yifan Wei, Liuji Chen, Liang Wang, Qiang Liu, and Shu Wu. Ex-fever: A dataset for multi-hop explainable fact verification. **arXiv preprint arXiv:2310.09754**, 2023.

[5] Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. Hover: A dataset for many-hop fact extraction and claim verification. **arXiv preprint arXiv:2011.03088**, 2020.

[6] Yuyu Zhang, Ping Nie, Arun Ramamurthy, and Le Song. Answering any-hop open-domain questions with iterative document reranking. In **Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval**, pp. 481–490, 2021.

[7] Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wentau Yih, Sebastian Riedel, Douwe Kiela, et al. Answering complex open-domain questions with multi-hop dense retrieval. **arXiv preprint arXiv:2009.12756**, 2020.

[8] Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. Proofwriter: Generating implications, proofs, and abductive statements over natural language. **arXiv preprint arXiv:2012.13048**, 2020.

[9] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **The Journal of Machine Learning Research**, Vol. 21, No. 1, pp. 5485–5551, 2020.

[10] Kaiyu Yang, Jia Deng, and Danqi Chen. Generating natural language proofs with verifier-guided search. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 89–105, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[11] Danilo Neves Ribeiro, Shen Wang, Xiaofei Ma, Rui Dong, Xiaokai Wei, Henry Zhu, Xinchi Chen, Zhiheng Huang, Peng Xu, Andrew O. Arnold, and Dan Roth. Entailment tree explanations via iterative retrieval-generation reasoner. In **NAACL-HLT**, 2022.

[12] Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. Rlet: A reinforcement learning based approach for explainable qa with entailment trees. **arXiv preprint arXiv:2210.17095**, 2022.

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

[14] Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. Entailer: Answering questions with faithful and truthful chains of reasoning. In **Conference on Empirical Methods in Natural Language Processing**, 2022.