LoraMap: Harnessing the Power of LoRA Connections

Anonymous ACL submission

Abstract

The advancement of Large Language Models (LLMs) benefit from fact-checking to mitigate hallucination and parameter-efficient techniques such as Low-rank adaptations (LoRA) to overcome enormous computational overhead. While some studies have explored the parallel integration of multiple LoRAs, these approaches need attention to the connections between them. This paper investigates methods to establish connections among multiple LoRAs inspired by the information processing behavior of the human brain. We create three reasoning datasets tailored to fact-checking and fine-tune individual LoRAs, allowing them to view and reason from diverse perspectives. Then, we explore strategies for allocating these reasoning LoRAs and introduce LoraMap, an approach to map connections between them. The results on the fact-checking task demonstrate the superior performance of LoraMap over LoraHub, an existing LoRA composition method. LoraMap also achieves higher performance with significantly fewer parameters than LoraConcat, which concatenates LoRAs and further fine-tunes them.

1 Introduction

With the rapid progress in research leveraging Large Language Models (LLMs) such as GPT-4 (OpenAI, 2023), PaLM (Chowdhery et al., 2023), LLaMA (Touvron et al., 2023), and Flan-T5 (Chung et al., 2022) in various natural language processing tasks, several challenges have also emerged. The model can pose a significant risk to reliability and trustworthiness due to the issue of generating false information, known as hallucination (Ji et al., 2023). One way to alleviate this problem is using fact-checking to verify LLM outputs or stand-alone claims (Gupta et al., 2022; Chamoun et al., 2023).

As in Figure 1, a fact-checking process classifies a claim into true, false, or more sophisticated labels based on textual evidence such as Wikipedia passages, news articles, and other relevant documents (Thorne et al., 2018; Guo et al., 2022). In biomedical and health domains, serious problems can arise when people perceive false information as truth, highlighting the importance of fact-checking. Accordingly, many studies have been explored, resulting in the development of datasets: SciFact (Wadden et al., 2020), PubHealth (Kotonya and Toni, 2020), COVID-Fact (Saakyan et al., 2021), and HealthVer (Sarrouti et al., 2021). This paper focuses on the COVID-Fact dataset, which covers fact-checking related to the COVID-19 pandemic.

Another challenge is that fine-tuning LLMs requires enormous computational resources and memory. Parameter-efficient fine-tuning techniques can address this issue, especially Low-rank adaptations (LoRA) (Hu et al., 2021). Furthermore, some studies have investigated the parallel integration of multiple LoRAs by learning weights for each LoRA and computing the weighted sum (Pfeiffer et al., 2021; Huang et al., 2023). The parallel linear sum of LoRAs may weaken the pivotal LoRA and overlook the connections between the knowledge embedded in each LoRA.

This paper investigates the methods of establishing connections among LoRAs to exchange their specialized insights as an alternative to parallel integration. Our main contributions are as follows:

- We create three reasoning datasets tailored to fact-checking and fine-tune LoRA for each dataset, allowing them to infer from different perspectives.
- We investigate how to connect these reasoning LoRAs and introduce LoraMap, which learns to map their connections.
- The results on the COVID-Fact dataset demonstrate that LoraMap exhibits superior performance than LoraHub, and also slightly outperforms LoraConcat even with significantly fewer parameters.

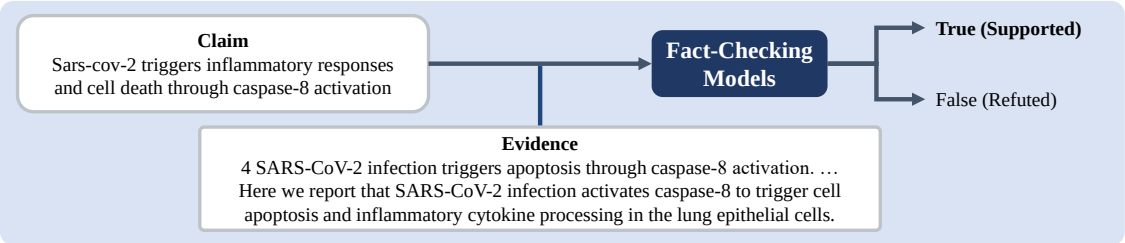

Figure 1: A fact-checking task classifies a claim as true or false based on the corresponding evidence.

2 Methods

2.1 Reasoning Dataset Generation

We hypothesize that identifying contrasting or common factors between the claim sentence and its corresponding evidence text can help the fact-checking model. Hence, we customize the three reasoning tasks for fact-checking: DifferenceCoT, EntityCoT, and CorrectClaim.

DifferenceCoT is a task that generates a text that details the contextual differences between claim and evidence, such as relation, topic, and level of detail.

EntityCoT is a task that extracts synonymous biomedical entities that appear simultaneously in the claim sentence and the evidence text.

CorrectClaim is a task that revises a given claim sentence based on the evidence.

Next, we construct datasets for these three tasks as follows. First, we extract 2,550 claim-evidence pairs from the COVID-Fact dataset and split them into 2,036 training instances, 258 development instances, and 256 test instances, as indicated in Table 1. The extraction process involves randomly selecting two of those claims, one true and the other false, for each piece of evidence. For DifferenceCoT and EntityCoT, we employ Chain-of-Thought (CoT) prompting (Wei et al., 2022) with the GPT-4 API to generate output text. On the contrary, the CorrectClaim dataset is based on the COVID-Fact dataset consisting of evidence and its corresponding claims with a veracity label. To generate the dataset for CorrectClaim, we extract claim-evidence pairs as inputs and assign them to a true output claim. The truthfulness of the output is guaranteed as follows: if the input claim is true, the output is the same as the input, and if the input claim is false, the output is a true claim from the given evidence. Figure 2 shows an example of instructions to a generative model, an input claim and evidence(context), and the generated output text.

Split	True	False	Total
Train	1,018	1,018	2,036
Dev	129	129	258
Test	128	128	256

Table 1: The statistics of the reasoning datasets DifferenceCoT, EntityCoT, and CorrectClaim.

2.2 Fine-tuning Reasoning LoRAs

The next step is to fine-tune LoRAs for each task. We use Flan-T5-large[1] as the base model and equip the lightweight module LoRA into all transformer attention layers. Specifically, as shown in Figure 2, LoRA operates within the query and value parts of the encoder self-attention, decoder self-attention, and encoder-decoder attention layers. For each task $t \in \{1, 2, 3\}$, LoRA consists of a weight matrix $A_t \in \mathbb{R}^{d \times r}$ for down-projection of features to a smaller dimension r, and a weight matrix $B_t \in \mathbb{R}^{r \times d}$ for up-projection to the original dimension d. By freezing the weights of the base model and training only the weights of LoRA, training requires approximately 4M parameters out of 797M parameters.

2.3 Connecting Reasoning LoRAs

The final step is to investigate methods for allocating and connecting the reasoning LoRAs, namely LoraHub, LoraConcat, and LoraMap. Figure 3 illustrates the differences among the methods.

LoraHub computes the weighted sum to generate $\hat{A}_t \in \mathbb{R}^{d \times r}$ and $\hat{B}_t \in \mathbb{R}^{r \times d}$. This framework freezes all A and B matrices and learns only the coefficients for each LoRA using a gradient-free approach. Our LoraHub follows the original LoraHub setting[2] and loads three reasoning LoRAs along with the 20 LoRA modules used by LoraHub.

LoraConcat concatenates the matrices A_t and B_t of the three reasoning LoRAs to produce $A_{cat} = \bigoplus_{t=1}^{3} A_t \in \mathbb{R}^{d \times 3r}$ and $B_{cat} = \bigoplus_{t=1}^{3} B_t \in \mathbb{R}^{3r \times d}$,

[1] https://huggingface.co/google/Flan-T5-large
[2] https://github.com/sail-sg/lorahub

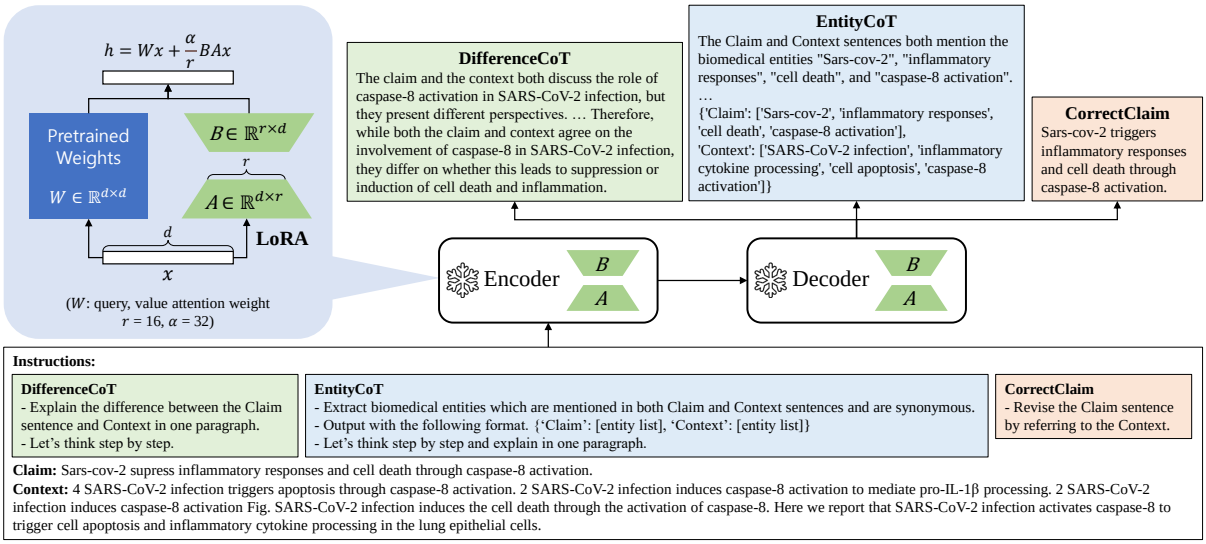

Figure 2: The LoRA exists in the query and value parts of all transformer attention layers and consists of A and B weight matrices. See Appendix B for the full generated output text.

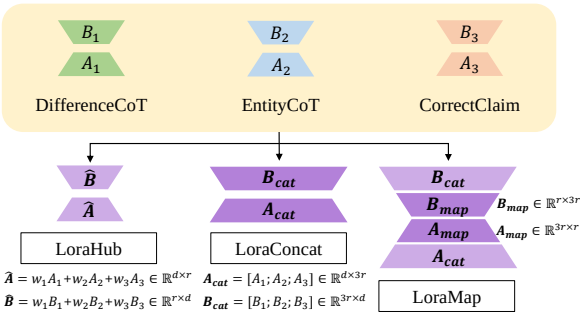

Figure 3: The comparison of LoraHub, LoraConcat, and LoraMap. Dark purple indicates trainable weights and light purple represents fixed weights.

respectively. We then fine-tune the A_{cat} and B_{cat} matrices targeting the COVID-Fact dataset. LoraMap not only concatenates the three reasoning LoRAs into A_{cat} and B_{cat} but also insert the trainable matrices $A_{map} \in \mathbb{R}^{3r\times r}$ and $B_{map} \in \mathbb{R}^{r\times 3r}$ between them. LoraMap freezes LoRAs that maintain specialized reasoning capabilities and learns the connection maps between them by fine-tuning only A_{map} and B_{map}. We intend to establish connections between multiple LoRAs directly by LoraConcat and indirectly by LoraMap.

3 Experimental Results

3.1 Reasoning LoRAs

We independently finetune DifferenceCoT LoRA, EntityCoT LoRA, and CorrectClaim LoRA inserted in the Flan-T5-large model. Table 2 shows the results of three reasoning LoRAs using BLEU (Papineni et al., 2002), ROUGE (Lin, 2004; Lin and Och, 2004), and METEOR (Banerjee and Lavie, 2005) scores as lexical overlap-based metrics, and BERTscore (Zhang et al., 2019) with the Longformer-base model (Beltagy et al., 2020) as semantic embedding-based metrics. In the zero-shot setting, the base model performs reasoning tasks without fine-tuning, resulting in poor scores. Fine-tuning LoRA on each reasoning dataset significantly increases the scores of all metrics. Revising a claim is easier than capturing differences or identifying synonymous entities, so the CorrectClaim scores are considerably higher than other tasks. When fine-tuning the three reasoning LoRAs, the experimental settings are identical, with a fixed seed 42 to ensure reproducibility. See Appendix A for details.

3.2 Connecting LoRAs for Fact-checking

We conduct experiments integrating multiple reasoning LoRAs on the COVID-Fact dataset. Given the prompt "*What is the class of the Claim by referring to the Context? Choose only from TRUE or FALSE.*" with claim and context, the output should be "*The claim is TRUE/FALSE*".

Table 3 presents the performance on the COVID-Fact test dataset, including macro-precision, macro-recall, and macro-f1 scores. In the zero-shot setting, using the GPT-4 API with Chain-of-Thought prompting yields an f1 score of 0.6959, indicating modest performance, and Flan-T5-large predominantly predicted TRUE with an f1 score of 0.5453.

Base model	Reasoning LoRA	Setting	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	METEOR	BERTscore
Flan-T5-large	DifferenceCoT	Zero-shot	0.0023	0.2173	0.1326	0.1815	0.2011	0.1047	0.8563
		LoRA finetuning	**0.3588**	**0.6676**	**0.4206**	**0.5045**	**0.6310**	**0.5255**	**0.9275**
	EntityCoT	Zero-shot	0	0.0539	0.0201	0.0533	0.0526	0.0289	0.7997
		LoRA finetuning	**0.3885**	**0.6755**	**0.4533**	**0.5548**	**0.6397**	**0.5969**	**0.9240**
	CorrectClaim	Zero-shot	0.3636	0.6839	0.5714	0.6618	0.6636	0.6591	0.9349
		LoRA finetuning	**0.9257**	**0.9722**	**0.9437**	**0.9721**	**0.9721**	**0.9682**	**0.9944**

Table 2: The evaluation results on three reasoning test datasets. The bold text represents the best result.

Model	Reasoning LoRA	Fact-checking setting	# Training instances	Macro-precision	Macro-recall	Macro-f1
GPT-4	—	—	0	0.7426	0.7070	0.6959
Flan-T5-large	—	—	0	0.7819	0.6133	0.5453
	base20 + DifferenceCoT + EntityCoT + ClaimCorrection	LoraHub	50	**0.6833**	**0.6797**	**0.6781**
			200	**0.6833**	**0.6797**	**0.6781**
			2,036*	0.6667	0.6562	0.6508
	DifferenceCoT + EntityCoT + ClaimCorrection	LoraConcat (14M)	100	0.7970	0.7930	0.7923
			1,000	0.8362	0.8125	0.8091
			2,036*	**0.8417**	**0.8203**	**0.8175**
	DifferenceCoT + EntityCoT + ClaimCorrection	LoraMap (0.22M)	100	0.7179	0.6328	0.5931
			1,000	**0.8347**	**0.8281**	**0.8273**
			2,036*	**0.8347**	**0.8281**	**0.8273**

Table 3: The evaluation results on the COVID-Fact test dataset. In the fact-checking settings, the value in parenthesis indicates the number of trainable parameters. The bold text represents the best result.
* is the size of all the training data.

The key result is a comparison of multiple LoRAs connecting methods: LoraHub, LoraConcat, and LoraMap. We experiment with various training instances, and Table 3 presents the best result among 10-shot, 20-shot, 50-shot, and 100-shot, the best result among 200-shot, 500-shot, and 1000-shot, and the result when using the entire dataset. LoraHub achieves the highest f1-score of 0.6781 at 50-shot and 200-shot, and its performance does not increase as the number of training data increases. Although training LoraHub with less than 100 examples is feasible, performance is poor in these settings. In contrast, LoraConcat and LoraMap generally demonstrate improved f1-scores as training instances increase. Notably, LoraConcat yields the best f1-score of 0.8175 when using all instances, and LoraMap achieves the highest f1-score of 0.8273 when using 1000-shot and all examples. Comparing LoraConcat and LoraMap, the scores are very similar, but LoraMap exhibits marginally superior performance with significantly fewer parameters (0.22M) than LoraConcat (14M).

4 Discussion

The experimental findings highlight the significance of the connection and allocation strategies of multiple reasoning LoRAs. After learning with the COVID-Fact dataset, LoraHub shows coefficients, which is the impact of each LoRA module. The coefficients for the three reasoning LoRAs are all close to 0.5, four out of the 20 base modules also exhibiting 0.5, mostly trained for question-answering tasks, and the remaining 16 show values close to zero or negative. The coefficients confirm that our reasoning LoRAs play an important role in fact-checking.

LoraConcat may lose reasoning capability as the concatenated LoRA matrices undergo further fine-tuning. To address this, we design LoraMap, which preserves these matrices in their original states and learns only the connection mappings among LoRAs to facilitate decision-making from diverse reasoning perspectives. This approach is inspired by the way the human brain processes information. As each brain region possesses different knowledge and functionalities, establishing interconnections among them would be important.

5 Conclusion

This paper investigates methods to establish connections among multiple reasoning LoRAs. We generate three reasoning datasets and fine-tune individual LoRAs to enable inference from different perspectives. Subsequently, we introduce LoraMap, an approach to learning the connection map between them. Our LoraMap outperforms LoraHub and LoraConcat, even with significantly fewer parameters. We anticipate that this paper will pave the way for novel approaches in mapping and designing connections among LoRAs.

6 Limitations

For each piece of evidence, there are true and false claims in the COVID-Fact dataset, so we automatically generate the CorrectClaim dataset. However, to apply this to other fact-checking datasets, researchers should consider the CoT prompting with GPT-4, similar to DifferenceCoT and EntityCoT. Additionally, it is essential to establish a method for assessing the quality of GPT-4 reasoning. Our model is not suitable for cases where only claims are present without evidence. In this case, appropriate evidence should be searched and provided. Making integrated judgments regarding multiple pieces of evidence is also impossible. Finally, examining LoraConcat and LoraMap on various open-source LLMs and other fact-checking datasets in the biomedical and health domains is necessary.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, Prague, Czech Republic. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *Computing Research Repository*, arXiv:2004.05150. Version 2.

Eric Chamoun, Marzieh Saeidi, and Andreas Vlachos. 2023. Automated fact-checking in dialogue: Are specialized models needed? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16009–16020, Singapore. Association for Computational Linguistics.

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. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Computing Research Repository*, arXiv:2210.11416.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Prakhar Gupta, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Dialfact: A benchmark for fact-checking in dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3785–3801, Dublin, Ireland. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, Online. Association for Computational Linguistics.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. volume arXiv:2307.13269.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out, Association for Computational Linguistics*, pages 74–81.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using

longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *Computing Research Repository*, arXiv:2303.08774.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. pages 487–503, Online. Association for Computational Linguistics.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. Covid-fact: Fact extraction and verification of real-world claims on covid-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1:Long Papers)*, pages 2116–2129, Online. Association for Computational Linguistics.

Mourad Sarrouti, Asma Ben Abacha, Yassine Mrabet, and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In *In Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3499–3512, Punta Cana, Dominican Republic. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Computing Research Repository*, arXiv:2302.13971.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy L. Wang, Madeleine V. Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. volume 35, pages 24824–24837, New Orleans, Louisiana, United States of America. NeurIPS.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *International Conference on Learning Representations*.

A Experimental Details

A.1 Details of Fine-tuning Reasoning LoRAs

When fine-tuning the three reasoning LoRAs, the experimental settings are identical, with a fixed seed 42 to ensure reproducibility. The maximum source length of the base model is set to 1200, and the maximum target length is constrained to 512. The LoRA rank parameter is configured to 16, with α set to 32. Throughout the 20 epochs of training, we employ early stopping with the patience of 3, selecting the epoch yielding the best ROUGE-Lsum score on the development set. The learning rate is $1e-3$, and we utilize the adafactor optimizer coupled with a cosine scheduler. Our setup involves two RTX 3090 GPUs, with a batch size per device of 1 and a gradient accumulation step of 8. Training takes 6 hours and 50 minutes for DifferenceCoT, 8 hours and 8 minutes for EntityCoT, and 3 hours and 46 minutes for CorrectClaim. Inferencing on the test dataset requires 28 minutes for DifferenceCoT, 29 minutes for EntityCoT and 4 minutes for CorrectClaim. CorrectClaim has a shorter output than other tasks, taking less time.

A.2 Details of Fine-tuning for Fact-checking

The experimental settings of fine-tuning LoraConcat and LoraMap on COVID-Fact are identical to that of fine-tuning the reasoning LoRAs, except that the gradient accumulation step is set to 4. LoraHub demonstrates faster training times, about 2 minutes for 50-shot and 1 hour and 44 minutes for using all the training dataset. However, its performance needs to catch up to that of LoraConcat, which requires 5 hours and 7 minutes, and LoraMap, which takes 4 hours and 14 minutes. Inferencing on the COVID-Fact test dataset takes less than 1 minute for LoraHub and less than 2 minutes for LoraConcat and LoraMap.

B Examples of datasets

Figure 4, 5, and 6 show an example of DifferenceCoT, EntityCoT, and CorrectClaim, respec-

DifferenceCoT

Instructions:
- Explain the difference between the Claim sentence and Context in one paragraph.
- Let's think step by step.

Claim: Sars-cov-2 suppress inflammatory responses and cell death through caspase-8 activation.

Context: 4 SARS-CoV-2 infection triggers apoptosis through caspase-8 activation. 2 SARS-CoV-2 infection induces caspase-8 activation to mediate pro-IL-1β processing. 2 SARS-CoV-2 infection induces caspase-8 activation Fig. SARS-CoV-2 infection induces the cell death through the activation of caspase-8. Here we report that SARS-CoV-2 infection activates caspase-8 to trigger cell apoptosis and inflammatory cytokine processing in the lung epithelial cells.

Explanation:

generate reasoning dataset using GPT-4

The claim and the context both discuss the role of caspase-8 activation in SARS-CoV-2 infection, but they present different perspectives. The claim suggests that SARS-CoV-2 suppresses inflammatory responses and cell death through caspase-8 activation. This implies that the virus uses caspase-8 activation as a mechanism to inhibit the body's immune response and prevent cell death, potentially aiding its survival and propagation. On the other hand, the context provides information that SARS-CoV-2 infection triggers apoptosis (cell death) and inflammatory cytokine processing through caspase-8 activation. This suggests that the virus's interaction with caspase-8 leads to cell death and inflammation, which are typical responses of the body's immune system to infection. Therefore, while both the claim and context agree on the involvement of caspase-8 in SARS-CoV-2 infection, they differ on whether this leads to suppression or induction of cell death and inflammation.

Figure 4: An example of DifferenceCoT dataset.

tively. Figure 7 shows an example of fact-checking prompt when using zero-shot CoT with GPT-4 API. Figure 8 shows an example of fact-checking dataset for fine-tuning Flan-T5 model.

EntityCoT

Instructions:
- Extract biomedical entities which are mentioned in both Claim and Context sentences and are synonymous.
- Output with the following format.
{'Claim': [entity list], 'Context': [entity list]}
- Let's think step by step and explain in one paragraph.
Claim: Sars-cov-2 suppress inflammatory responses and cell death through caspase-8 activation.
Context: 4 SARS-CoV-2 infection triggers apoptosis through caspase-8 activation. 2 SARS-CoV-2 infection induces caspase-8 activation to mediate pro-IL-1β processing. 2 SARS-CoV-2 infection induces caspase-8 activation Fig. SARS-CoV-2 infection induces the cell death through the activation of caspase-8. Here we report that SARS-CoV-2 infection activates caspase-8 to trigger cell apoptosis and inflammatory cytokine processing in the lung epithelial cells.
Explanation:

generate reasoning dataset using GPT-4

The Claim and Context sentences both mention the biomedical entities "Sars-cov-2", "inflammatory responses", "cell death", and "caspase-8 activation". In the context, "SARS-CoV-2 infection" is synonymous with "Sars-cov-2" from the claim. Similarly, "cell apoptosis" and "cell death" are synonymous terms. "Inflammatory cytokine processing" and "inflammatory responses" are also synonymous. Lastly, "caspase-8 activation" is mentioned in both sentences with the same meaning. Therefore, the extracted entities are "Sars-cov-2", "inflammatory responses", "cell death", and "caspase-8 activation".
The output in json format would be:
{
'Claim': ['Sars-cov-2', 'inflammatory responses', 'cell death', 'caspase-8 activation'],
'Context': ['SARS-CoV-2 infection', 'inflammatory cytokine processing', 'cell apoptosis', 'caspase-8 activation']
}

Figure 5: An example of EntityCoT dataset.

CorrectClaim

Instructions:
- Revise the Claim sentence by referring to the Context.
Claim: Sars-cov-2 suppress inflammatory responses and cell death through caspase-8 activation.
Context: 4 SARS-CoV-2 infection triggers apoptosis through caspase-8 activation. 2 SARS-CoV-2 infection induces caspase-8 activation to mediate pro-IL-1β processing. 2 SARS-CoV-2 infection induces caspase-8 activation Fig. SARS-CoV-2 infection induces the cell death through the activation of caspase-8. Here we report that SARS-CoV-2 infection activates caspase-8 to trigger cell apoptosis and inflammatory cytokine processing in the lung epithelial cells.
Revised Claim:

generate reasoning dataset using GPT-4

Sars-cov-2 triggers inflammatory responses and cell death through caspase-8 activation.

Figure 6: An example of CorrectClaim dataset.

Fact-checking using GPT-4 (zero-shot CoT)

Instructions:
You are a helpful assistant with biomedical expertise.
Instructions:
- Classify the Claim sentence only as 'TRUE' or 'FALSE' by referring to the Context. \
- Let's think step by step to explain why in one paragraph.
Claim: Sars-cov-2 suppress inflammatory responses and cell death through caspase-8 activation.
Context: 4 SARS-CoV-2 infection triggers apoptosis through caspase-8 activation. 2 SARS-CoV-2 infection induces caspase-8 activation to mediate pro-IL-1β processing. 2 SARS-CoV-2 infection induces caspase-8 activation Fig. SARS-CoV-2 infection induces the cell death through the activation of caspase-8. Here we report that SARS-CoV-2 infection activates caspase-8 to trigger cell apoptosis and inflammatory cytokine processing in the lung epithelial cells.
Explanation:

The claim that Sars-cov-2 triggers inflammatory responses and cell death through caspase-8 activation is TRUE. The context provided clearly states that SARS-CoV-2 infection activates caspase-8, which in turn triggers cell apoptosis (cell death) and inflammatory cytokine processing in lung epithelial cells. The context also mentions that SARS-CoV-2 infection induces pro-IL-1β processing, which is a part of the inflammatory response. Therefore, the claim is consistent with the information provided in the context.

Figure 7: An example of fact-checking prompt when using zero-shot CoT with GPT-4.

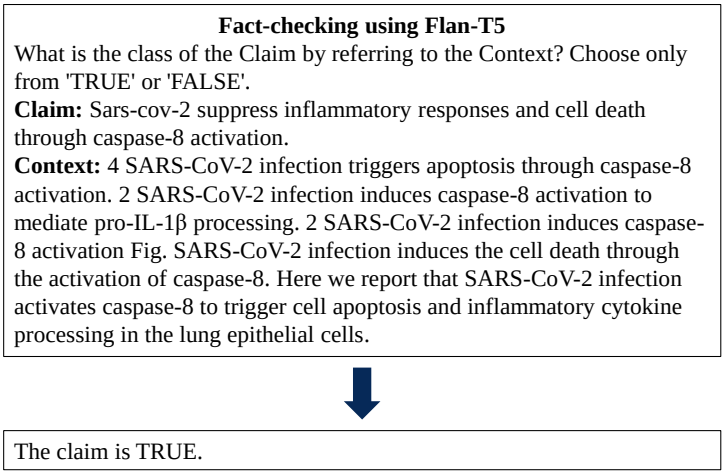

Figure 8: An example of fact-checking dataset for fine-tuning Flan-T5 model.