Improving Chain-of-Thought Reasoning for Visual Question Answering

Eric Wang, Phuc Duong, Sophia Kang
Department of Computer Science
Yale University
New Haven, CT 06511
eric.y.wang@yale.edu, phuc.duong@yale.edu, sophia.kang@yale.edu

Abstract

The Visual Question Answering (VQA) task is difficult for vision-language and language models because it requires answering questions that involve not only multimodal inputs but also multiple reasoning steps. We introduce 3 new systems for Chain-of-Thought (CoT) for VQA: (1) dynamic CoT, which generates sub-questions tailored to each image-question pair, (2) self-consistent CoT, which samples multiple reasoning chains and picks the most frequent answer, and (3) sequential CoT which feeds in the answer to a sub-question back into the model, before generating a new one to iteratively develop further sub-questions. We found that sequential CoT performed 10.59% better compared to basic CoT in BLIP-2+GPT4.1 and 36.6% better with ViLT+o4-mini. Analysis also shows that sequential prompting can help correct hallucinations and mitigate error propagation to some questions. Additionally, we also found that CoT prompting is explicitly better on questions that compare attributes of two different objects, as it isolates each attribute into simpler sub-questions the VQA model can answer more accurately. Our results show that iterative and dynamic reasoning with CoT can help improve multi-step VQA. Our code is available at https://github.com/phucd5/vqa-cot.

1 Project Motivation

Visual Question Answering (VQA) refers to the task of providing an accurate natural language answer given an image and a natural language question about the image [1]. The VQA task is difficult for vision-language and language models because complex visual inputs involve not only identifying the region in the image that contains the answer but also applying a series of reasoning abilities. This task becomes more complex when the evaluation dataset includes questions that require multi-step reasoning, as in the GQA dataset [4]. Because this dataset requires a multitude of complex skills related to visual question answering, such as multiple reasoning skills, spatial understanding, and multi-step inference, we turn to chain-of-thought to simplify the question-answering pipeline. Specifically, LLMs are used to generate and break down a main question into relevant subquestions to be answered step-by-step, and they are also used when aggregating the final answer from the subquestions. We test multiple of these pipelines and provide results as outlined in the sections below.

2 Related Work

One approach to tackling the visual question-answering task is by using annotated question-answer pairs. [11] used question-answer pairs annotated with detailed reasoning steps and intermediate bounding boxes highlighting key regions in the image for answering the questions. Other approaches have included interactions between visual content and natural language in an iterative step-by-step

Preprint. Under review.

reasoning manner with 3 steps: see, think, and confirm [3]. Some works used large language models as base question-answering models as in [8, 2]. In particular, [8] also used question-answer pairs that are augmented with annotated lectures and detailed explanations.

A different approach to deriving an answer for the visual question-answering task involves subquestion generation. [13] used a questioner-oracle-answerer structure involving a hierarchical recurrent encoder-decoder model, while [12] proposed using a VQA model alongside a Visual Question Generation model based on an encoder-decoder structure for generating questions. In the audio-visual question answering task, [6] leveraged an LLM to select relevant sub-questions from a sub-question bank and also used an LLM to gather the subquestion-answer pairs and generate the final answer while using an audio-visual model for answering the subquestions themselves.

3 Proposed Approach

While [6] introduced the Chain-of-Thought (CoT) reasoning framework to decompose complex questions for Audio-Visual Question-Answering (AVQA), their approach is limited by their reliance on a fixed set of predefined sub-questions. We propose three novel extensions to enhance reasoning for Visual Question-Answering (VQA): (1) dynamic generation of CoT sub-questions, eliminating dependence on static sub-question sets; (2) self-consistent CoT to improve robustness; and (3) sequential CoT to encourage incremental refinement of reasoning. We compare these systems to a baseline prompting strategy, which involves directly passing the complex visual question to the VQA model to retrieve the final answer. Figure 1 visualizes the overall design of each system.

The improved basic CoT system closely follows the strategy introduced in [6] for question decomposition and aggregation. The input question is passed into an LLM with a specialized prompt that asks the LLM to generate sub-questions. These sub-questions are then individually answered by the VQA model, and the sub-answers are aggregated by the LLM into the final answer to the complex question. Unlike the original CoT strategy in [6], which relies on choosing from a fixed, predefined set, this method leverages the LLM's generative capabilities to create sub-questions on-demand. This aims to make the system more versatile in answering a broad variety of visual questions.

The self-consistent CoT system draws from the work done by [14]. The self-consistency method comprises sampling from a diverse set of an LLM's reasoning paths and aggregating the final result by taking the answer with the highest frequency. This ensures that the system's generations across multiple samples are consistent with itself, leading to a more robust final answer.

The problem with the basic CoT system is that, since the sub-questions are generated all at once, sub-questions generated later are not able to improve upon the sub-questions before it. This suggests a need to explore an iterative sub-question generation strategy that we call sequential CoT. After the initial complex visual question is inputted into the LLM, only one sub-question is generated. The sub-question and its corresponding sub-answer are passed back into the LLM as context to produce the next sub-question. This process repeats for at least 2 sub-questions and, at most, k (5) sub-questions, with each subsequent sub-question becoming increasingly refined. The LLM is also prompted to stop asking sub-questions once it determines it has gathered all the necessary information to answer the main question accurately and minimize noise. Finally, all sub-question and answer pairs are aggregated as input into the LLM, and a final answer is produced.

3.1 Models

The vision-language models that we utilize are twofold: one for each model type in generation and one for the classification model. We examine BLIP-2 [7], a generative vision-language model and ViLT [5], a classification vision-language model. In addition, we use two types of LLMs, GPT 4.1 [9] and o4-mini [10]. We incorporate these two models into our work to compare the ability of the non-reasoning and reasoning LLMs to generate useful subquestions that can contribute to logically breaking down the larger question in the task.

3.2 Data preprocessing

We utilized the GQA: Visual Reasoning in the Real World dataset, which comprises of question-answer pairs pertaining to an image. Prior to training, we transformed the raw GQA JSON

into a flattened version containing only the fields we need (`question_id, question, answer, full_answer, and image_path`). We used stratified sampling by local group to ensure that the selected questions were representative of the various types and scenarios found in the full dataset for both training, evaluation, and validation.

We split preprocessing encoding into two groups, one for the classification model (ViLT) and one for the generative model (BLIP-2). For classification, we converted each answer into a one-hot vector over the full label set. Each answer is a one-word answer taken from the dataset. For generative preprocessing, we wrapped each question in a template (e.g., `"Question: {...} Answer:"`) and then masked each padding token that did not contribute to the loss. Instead of using the one-word answer, since our model is generative, we used the "full answer," which is a short sentence. This allows BLIP-2 to learn to generate complete answers in natural language rather than a single word.

3.3 Training

To train ViLT, we used an NVIDIA A100 80GB GPU for training and evaluation on the Yale High Performance Computing (HPC) clusters. We used an Intel Xeon Gold 6326 CPU with 36 CPU cores, with 128 GB of RAM available on the cluster, and 6 CPUs allocated per task. The pre-trained ViLT model is available on Hugging Face at `dandelin/vilt-b32-mlm`, and our fine-tuned version at `phucd/vilt-gqa-ft`. Similarly, the base BLIP-2 model can be found at `Salesforce/blip2-opt-2.7b`. We used 40,000 examples for training and 5,000 for validation.

Parameter	**Value**
Train Batch Size	16
Gradient Accumulation Steps	2
Total Train Batch Size	32
Learning Rate	5e-5
Optimizer	AdamW (betas=(0.9, 0.999), epsilon=1e-08)
Optimizer Args	No additional optimizer arguments
LR Scheduler Type	Linear
Number of Epochs	20
Eval Batch Size	8
Seed	42

Table 1: Hyperparameters for dandelin/vilt-b32-mlm

These parameters were chosen to ensure the models were trained in a timely manner, despite the resource constraints of Yale's HPC, while still achieving meaningful performance. Future work can be done to explore more extensive hyperparameter tuning to improve the vision model base accuracy.

Due to compute constraints, we were unable to train BLIP-2 for a significant duration, which impacted performance more severely than simply using the base model, so we opted to use the base model for evaluation. In addition, because pretrained ViLT's classifier head hasn't been trained on VQA labels and only returns placeholder classes, we could not evaluate the base model.

3.4 Evaluation

To evaluate our systems, we obtain a prediction via one of our four systems: direct, CoT, self-consistent CoT, or sequential CoT. We compare this with the ground-truth answer found in the GQA dataset for 250 question-image pairs. We opted to use the one-word answer rather than the full answer, so it simplifies comparison across models and systems and helps yield more consistent accuracy metrics. For each set of comparisons, we normalize both text by converting it to lowercase and stripping punctuation to ignore minor token-order differences.

In preliminary results, it was discovered that evaluation with compound nouns yields negative or unfair results. For example, the model is instructed to only generate one-word answers such as "stop" when the true answer is "stop sign." As a result, we allowed for single-word discrepancies to mitigate unfair penalization due to compound-noun labels. We consider a prediction correct when at least one word from either the prediction or the ground-truth answer appears as a substring of a word in the

VQA Model	LLM	Direct	CoT	Self-Consistent CoT	Sequential CoT
BLIP-2	None	0.428	—	—	—
	GPT 4.1	—	0.340	0.352	0.376
	o4-mini	—	0.340	0.336	0.368
ViLT	None	0.456	—	—	—
	GPT 4.1	—	0.292	0.3	0.336
	o4-mini	—	0.264	0.32	0.36

Table 2: Visual question answering evaluation.

other. From our analysis, this does not introduce any false-positive evaluation, as answers are always a single lexical item or fixed compound nouns.

4 Results

We evaluated each of the prompting strategies 3 with different configurations of VQA and LLM model types by measuring question-answering accuracy over 250 evaluation samples. The direct prompting strategy yielded the highest accuracy, scoring 0.428 for BLIP-2 and 0.456 for ViLT. BLIP-2 achieves higher accuracy than ViLT across all chain-of-thought (CoT) variants. For both models, self-consistent CoT provides a modest gain over basic CoT, while sequential CoT yields the best results. Notably, when pairing ViLT with OpenAI's o4-mini, self-consistent and sequential CoT improve accuracy by 21.2% (0.320 vs. 0.264) and 36.4% (0.360 vs. 0.264), respectively, compared to basic CoT.

5 Discussion

Our initial hypothesis was that all CoT methods would improve upon direct prompting for VQA accuracy. However, the results proved this hypothesis false. This could be attributed to the sub-par abilities of the underlying vision model. With direct prompting, both BLIP-2 and ViLT get the complex question wrong less than half of the time (0.428 and 0.456 respectively). After the VQA models answer multiple sub-questions, the noise (and potential conflicting conclusions) from the sub-question answers confuse the LLM at the final aggregation step, making the CoT systems produce more incorrect final answers than the direct answers.

For instance, consider the question in Figure 2, "On which side of the photo is the person, the left or the right?" Under direct prompting, BLIP-2 correctly answers "right." However, in the CoT pipeline, the BLIP-2 model answers the sub-question incorrectly, stating that there is no person in the photo. When the LLM (o4-mini) aggregates these sub-answers, it outputs "neither," illustrating how mistakes in the chain can cascade and undo any benefit of step-by-step reasoning. With a stronger model, the sub-answers may be far more reliable, and CoT could outperform direct prompting.

Although direct prompting methods were overall superior to the CoT methods across all 250 evaluation samples, the CoT methods were superior for handling certain question types. Looking at Figure 3 as an example, we found that CoT prompting is explicitly better on questions that compare attributes of two different objects. Decomposing the tasks into simpler binary questions allows the VQA model to attend to each object in isolation, improving the ability to get a correct final answer in aggregation. In contrast, direct prompting forces the VQA to identify both objects at the same time in one shot, leading to the model being misled by background context or missing key distinctions, leading to an incorrect answer.

One notable finding is that, although ViLT paired with o4-mini under basic CoT performs substantially worse than BLIP-2 with o4-mini (0.264 vs. 0.340), both self-consistent and sequential CoT narrow this gap, achieving performance comparable to BLIP-2. The lower basic CoT accuracy for ViLT is likely a reflection of its fewer parameters (compared to BLIP-2) and noise caused by o4-mini's diverse reasoning. This same "noise," however, is mitigated, or even harnessed, by the more complex CoT frameworks. Self-consistent CoT takes advantage of answer diversity through majority voting,

boosting accuracy to 0.320 (vs. 0.300 for ViLT+GPT 4.1), while sequential CoT's incremental reasoning makes use of the more diverse output from o4-mini to further refine each sub-question, raising performance to 0.360 (vs. 0.336 for ViLT+GPT 4.1).

5.1 Variations in CoT

Despite the overarching results, there are notable findings. One of these is that Sequential CoT shows significant over normal CoT. For example, when paired with BLIP-2, GPT-4.1 saw a 3.6% difference in accuracy from 34.0% to 37.6%. With BLIP-2+o4-mini, we also see a gain of 2.8% from 34.0% to 36.8%. Similarly ViLT+GPT-4.1 jumped from 29.2% to 33.6%, and with o4-mini increased from 26.4% to 36.0%. This shows that there is a benefit from a sequential approach when dealing with multi-step reasoning in VQA. Looking at Figure 4, with Sequential CoT, we can see it succeeded despite having several incorrect/confused intermediate sub-answers, like "yogurt." However, the LLM picked up on the hallucination by the VQA model due to the sequential nature of our prompting and asked other follow-up questions to clarify the relevant food attributes. By asking follow-up questions that relate to food type and shape afterward, the model was able to reason past the hallucination and identify the correct answer despite both direct prompting and CoT failing. Overall, Sequential CoT exhibits better performance by generating and answering each sub-question one at a time and then feeding the VQA model output back into the next step; the model can adjust to either avoid compound errors or validate the VQA response to get better performance than creating k sub-questions in one shot.

5.2 o4-mini vs GPT-4.1: Model Comparison

In our experiments, we used GPT 4.1 and o4-mini to see if a reasoning model would enhance our sub-question generation and be able to better aggregate the answers, leading to better performance. However, GPT-4.1 and o4-mini perform almost identically across all cases. For example, both models have 34% accuracy in standard CoT on BLIP-2. However, o4-mini performs worst in Sequential CoT at 36.8% compared to GPT 4.1 at 36.6%. Given that o4-mini is a reasoning model that takes longer to run and could be more expensive due to token generation, the benefits do not outweigh the cost and time of using o4-mini for sub-question generation and aggregation. However, future work can investigate if this still holds true when we have better base vision models.

5.3 Limitations and Future Works

Although we did identify some notable findings, our overall systems suffered from relatively poor base model accuracy. This caused errors to propagate through subquestions and degraded the final answers. Future work could improve the underlying vision-language model by trying a new model or training the model for longer. We were also constrained by budget when using OpenAI's API, so our evaluations were only done in one shot. Thus, we cannot account for the variations from run to run during our experiments or try a large number of different prompts to identify the best prompts for the task. Finally, our current pipelines apply the same CoT strategy to every question, regardless of its inherent complexity. This leads to resource waste on trivial questions but could also lead to performance degeneration, where trivial questions that the base vision model can answer correctly are answered incorrectly by our own pipelines. Future systems could implement a difficulty classifier, either using an LLM annotator or a small neural network that assesses the difficulty of the question, directing simple queries to the VQA model and more complex questions to our various CoT pipelines. This may not help improve accuracy but may save inference time and costs.

6 Conclusion

In this work, we tackle VQA by proposing three novel Chain-of-Thought (CoT) prompting strategies - dynamic, self-consistent, and sequential - to better support multistep reasoning processes for models with multimodal inputs, namely text and images. Among these, we find that sequential CoT performs the best, reduces hallucinations, and alleviates error propagation. It proves particularly effective for comparative questions by decomposing them into manageable sub-questions. Our findings underscore the potential of iterative and adaptive reasoning via CoT to enhance VQA performance, particularly when combining vision-language models and large language models in this task.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433, 2015.

[2] Rui Cao and Jing Jiang. Knowledge generation for zero-shot knowledge-based VQA. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 533–549, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[3] Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. Visual Chain-of-Thought Prompting for Knowledge-Based Visual Reasoning. volume 38, pages 1254–1262, Mar. 2024.

[4] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[5] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR, 18–24 Jul 2021.

[6] Guangyao Li, Henghui Du, and Di Hu. AVQA-CoT: When CoT Meets Question Answering in Audio-Visual Scenarios. In *CVPR Sight and Sound Workshops*, 2024.

[7] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[8] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *NeurIPS*, 2022.

[9] OpenAI. Introducing GPT-4.1, April 2025.

[10] OpenAI. Introducing OpenAI o3 and o4-mini, April 2025.

[11] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual CoT: Advancing Multi-Modal Language Models with a Comprehensive Dataset and Benchmark for Chain-of-Thought Reasoning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 8612–8642. Curran Associates, Inc., 2024.

[12] Kohei Uehara, Nan Duan, and Tatsuya Harada. Learning to ask informative sub-questions for visual question answering. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4680–4689, 2022.

[13] Ruonan Wang, Yuxi Qian, Fangxiang Feng, Xiaojie Wang, and Huixing Jiang. Co-VQA : Answering by Interactive Sub Question Sequence. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2396–2408, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[14] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models, 2023.

A Appendix

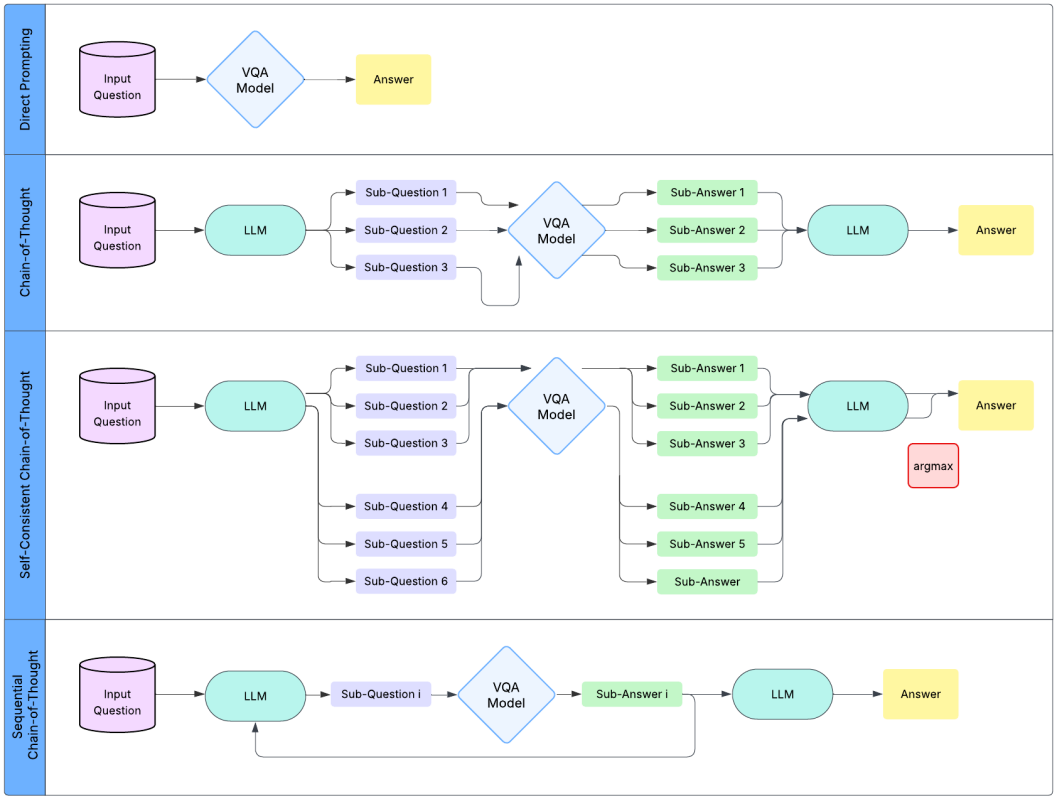

Figure 1: Chain-of-Thought Reasoning Systems for Visual Question-Answering

Qualitative CoT Failure Example

Main question:
On which side of the photo is the person, the left or the right?

Direct prompting:
right

CoT breakdown:

1. *Is a person visible in the photo?*
   Answer: no
2. *Is the person on the left side of the photo?*
   Answer: no

Aggregated CoT answer:
neither

Figure 2: Example showing failure of a Chain-of-Thought (CoT) approach, where the breakdown leads to a different answer than direct prompting using BLIP-2+o4-mini.

Chain-of-Thought (CoT) Example

Question: Is the palm made of the same material as the fence?

Sub-questions & Answers:

- Is the palm made of wood? **no**
- Is the fence made of wood? **no**
- Is the palm made of metal? **no**
- Is the fence made of metal? **yes**

Model Prediction: **no** (Correct)

(a) CoT reasoning yields the correct answer.

Direct Prompting Example

Question: Is the palm made of the same material as the fence?

Model Prediction: **yes** (Incorrect)

(b) Direct prompting fails on this question.

Figure 3: Comparison of Chain-of-Thought vs. Direct Prompting on a material comparison question. (ViLT+o4-mini)

Direct Prompting

Question:
What kind of food is to the left of the egg?

Prediction:
egg (Incorrect)

(a) Direct prompting fails.

CoT (o4-mini)

Question:
What kind of food is to the left of the egg?

Sub-questions & Answers:

- Is an egg visible? **yes**
- What food item is directly to the left of the egg? **egg**

Final Answer:
egg (Incorrect)

(b) CoT straightforward breakdown.

Sequential CoT (o4-mini)

Question: What kind of food is to the left of the egg?

Sub-questions & Answers:

- What is the color of the oval food item? **white**
- Is it bacon or sausage? **yogurt** (Wrong)
- Which type—dairy or meat—is it? **meat**
- Which shape—round or strip—is the meat? **round**
- Is it yogurt? **yes** (Wrong)

Final Answer: **sausage** (Correct)

(c) Sequential CoT solves despite errors.

Figure 4: Comparison on the question *"What kind of food is to the left of the egg?"* showing Direct prompting, CoT, and Sequential CoT. (VilT+o4-mini)

Reproducibility checklist

* Please make sure these points are addressed in your report submission
* Please copy this and replace the ☐ with a ✔ for the items that are addressed in your report/code submission
* Please complete this report, attach it to your final project report as the last page and then submit.

Model Description, algorithm, Mathematical Setting:

- ✔ Include a thorough explanation of the model/approach or the mathematical framework

Source Code Accessibility:

- ✔ Provide a link to the source code on github.
- ✔ Ensure the code is well-documented
- ✔ Ensure that the github repo has instructions for setting up the experimental environment.
- ✔ Clearly list all dependencies and external libraries used, along with their versions.

Computing Infrastructure:

- ✔ Detail the computing environment, including hardware (GPUs, CPUs) and software (operating system, machine learning frameworks) specifications used for your results.
    - (Example statement 1: the model was fine-tuned using a single T4 GPU on colab.
      Example statement 2: we ran inference of Llama 70B using 4 Nvidia A5000 GPUs)
- ✔ Mention any specific configurations or optimizations used.
    - (Example: We used a quantized version of Llama with int8.
      Example 2: We used the regular float32 representation.)

Dataset Description:

- ✔ Clearly describe the datasets used, including sources, preprocessing steps, and any modifications.
- ✔ If possible, provide links to the datasets or instructions on how to obtain them.

Hyperparameters and Tuning Process:

- ✔ Detail the hyperparameters used and the process for selecting them.
    - (Example: The model was fine-tuned using a batch size of 16, learning rate of 1e-5, and trained on 1000 steps with 100 steps of learning rate linear warmup with linear decay)

Evaluation Metrics and Statistical Methods:

✔ Clearly define the evaluation metrics and statistical methods used in assessing the model.

Experimental Results:

✔ Present a comprehensive set of results, including performance on test sets and/or any relevant validation sets.

✔ Include comparisons with baseline models and state-of-the-art, where applicable.

Limitations and future work:

✔ Include a discussion of the limitations of your approach and potential areas for future work.