Scientific Reasoning: Assessment of Multimodal Generative LLMs

Florian Dreyer (3667877) [*1] Ekaterina Kolos (3688474) [*2] Daria Matiash (3668740) [*2]

Abstract

This project assesses the capabilities of pretrained multimodal LLMs to perform scientific reasoning tasks on multimodal data. We further explore how prompt tuning (soft prompting) and prompt engineering techniques can improve performance. We then attempt to distill knowledge to a small LLM.

1. Introduction

SECTION NEEDS UPDATE Scientific machine reasoning is the application of AI techniques to simulate scientific reasoning processes, such as data interpretation, hypothesis generation, and causal inference. It facilitates scientific inquiry and discovery by enabling machines to independently analyze scientific data, generate hypotheses, and make predictions. For our project, we intend to compare different approaches around generative large language models to generate solutions for school science problems. On an existing benchmark with human solutions, we are planning to compare: a pure LLM with zero-shot and few-shot prompting (and possibly some more fine-grained prompting techniques), an agentic approach with ability to query external resources like textbooks in physics, maths etc, and soft-prompting techniques. We then plan to attempt knowledge distillation to a smaller LLM.

2. Motivation

TODO

[*]Equal contribution [1]Institut fuer KI [2]Institut fuer Maschinelle Sprachverarbeitung. Correspondence to: Florian Dreyer <c.vvvvv@googol.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

3. Background and Related Work

Prompting techniques A prompt is a command a user gives to a generative LLM used to guide its output for a downstream task, which contains several of the following: a role (persona) the LLM has to follow (e.g. "You are a helpful assistant"), a precise task description, examples (exemplars) for analogous learning (few-shot prompting), requirements on the process of reasoning, style instructions, output format requirements, additional information for in-context learning, emotion prompting components (highlighting why the task or a particular requirement is important). Prompts can be combined into sequences or graphs (with complex branching and parallelism). Of particular interest are small requirements on the reasoning process that can significantly improve performance when added to the prompt, such as "Let's think step by step" (?). Chain-of-thought prompts improve reasoning capabilities by asking the model to speak its process of thinking out loud, with variations including self-ask, step-back prompting, thread-of-thought, least-to-most prompting (decomposing then solving), plan-and-solve prompting, tree-of-thought, recursion-of-thought and many more (?). Soft prompting is an alternative to fine-tuning where the model is frozen while only a small number of "soft" prompt parameters are trained (?). These parameters are used to guide the model in the right direction.

Agents An important step forward in using LLMs are agent-based architectures (?) (?). They allow the LLM to act as an intelligent being capable of planning a solution of a complex task, while resorting to external resources on the way to acquire in-context the knowledge it does not have from pre-training. The retrieved information, results of invoking tools, such as calculators, and of interactions with the environment are appended to the prompt (or, in a more advanced scenario, to the summarized memory (?)). Optionally, a reflexion step (explicit reasoning on all the accumulated information) is added before allowing further generation of the final response.

Knowledge distillation (KD) allows to obtain smaller models capable of successfully following the behavior of larger teacher models. This is particularly useful for privacy reasons (mobile AI apps) and when access to large models in the cloud is not guaranteed (e.g. in cars). The teacher model can be used to intelligently select examples on which the student model is trained (dataset distillation) (?), or provide negative samples to show the student what incorrect answers or reasoning paths it should avoid to improve task accuracy (?) (c.f. contrastive CoT). Training small models on a CoT reasoning path of a larger model was also shown to be a way to obtain a small student model replicating reasoning capabilities of teacher on downstream tasks (?), which is close to response-based KD where the student model mimics the output of the teacher. Alternatively, with feature-based KD the knowledge from specific layers of the teacher model is distilled into the student model (?), while the student model's structure may be a quantized, simplified, or condensed version of the teacher's architecture (?).

In our project, we plan to develop an extended version of CoT knowledge distillation, where both the reasoning paths and the essence of the external knowledge the teacher agent has retrieved for the tasks are acting as training data for the student model. This way the teacher agent "defines" the "protocols" of solving complex tasks, while the student model tries to "follow" the learnt "instructions" and memorize important information for similar tasks. If we succeed, such small models could be a valuable asset in practice in enterprise environments, to encode important protocols and parts of project documentation, replacing expensive intelligent agents for specific tasks.

4. Methodology

We postulate the following research questions: RQ1: Does Soft Prompting improve the performance of the LLM for scientific reasoning? RQ2: Can a teacher model's knowledge be distilled into a student model's soft prompt?

RQ1 Our soft prompting approach relies on prefix soft prompts as introduced in [TODO: cite Li 2021], that can be used both for encoder-decoder and for autoregressive architectures. The models we test here are autoregressive.

5. Dataset

This study is based on the ScienceQA dataset (?). The dataset includes a variety of science-related multimodal multiple-choice questions together with annotations of the answers that provide relevant lectures and explanations in Nature Science, Language Science and Social Science.

For some datapoints, the image data is missing, while the answer can be deduced from text alone (e.g. questions on 'subject': 'language science', 'topic': 'figurative-language', 'skill': 'Interpret figures of speech' (e.g. irony recognition).

For other datapoints, however, the image data can be missing, although required to make an answer.

Apart from the image field, missing values can occur in lecture and solution fields. The statistics in table 1 give an overview on the amount of problematic datapoints.

Finally, hint can be missing, e.g. the text-only task 367 on subject language science: "What is the source of allusion in the sentence below? Edward picked up his pace on the trail as his spidey sense began to tingle" with choices ["Italian history", "a comic book"], or the text-only task 3 on the topic of capitalization: "Which correctly shows the title of a play?" with choices ["A breath of Fresh Air", "A Breath of Fresh Air"].

missing	frequency
solution	9.3%
lecture	15.7%
solution and/or lecture	24.2%
image	50.6%

Table 1. Defect datapoints in dataset

Retained data As 50.6% of the datapoints are missing an image, we decide to process both text-only and text+image datapoints similarly with multimodal models, attaching the image to the prompt if it is available. For now, we will only work with datapoints with non-empty solution AND with non-empty lecture. All the other datapoints are disregarded, which results in 3216 datapoints kept in validation data.

6. Experiments

6.1. Setup

We start by zero-shot benchmarking existing multimodal models in four simple prompt settings, partly repeating with the official evaluation suite:

1. question - choices - hint - image - task
2. question - choices - hint - image - task + lecture

3. question - choices - hint - image - task + lecture + solution

4. question - choices - hint - image - task + solution

For the first two the model has to output answer - solution, for the third - answer (the easiest set up meant to ensure the model can deduce the correct answer from an already provided correct solution).

We evaluate: OpenAI, Gemini, MistralAI, as well as small local models.

6.2. Metrics

6.2.1. Accuracy

The highest performance was observed in settings that included solutions, indicating that the models were generally capable of extracting relevant information effectively. Among these, settings combining both lecture and solution information typically yielded slightly better results than those with solutions alone. 'GPT-4' consistently received highest ratings, especially in scenarios in which lecture and/or solution information was included. The Gemini family of models showed better robustness in many scenarios and came in second overall. Notably, Gemini models achieved an average accuracy advantage of 5% over GPT-family models in the "pure" task context (without lecture or solution information). The 'LLaVAModel1.5-7b', on the other hand, demonstrated the lowest average accuracy scores, particularly in environments without lecture and solution data.

6.2.2. Cosine similarity

6.2.3. BLEU-1

Across settings, higher BLEU-1 scores were observed in settings 3 and 4 (those incorporating solutions and/or lecture information), indicating that models performed better with richer contextual data. The 'GPT-4o-mini' model demonstrated superior performance, surpassing 'GPT-4', in achieving the highest results. Lower scores in settings 1 and 2 suggest that models, particularly 'LLaVAModel1.5-7b', may have had trouble correctly aligning their outputs with reference words when given less auxiliary information, as indicated by lower scores in settings 1 and 2.

6.2.4. BLEU-4

Scores were consistently higher in settings 3 and 4, particularly for models, demonstrating their ability to produce more contextually precise outputs when both lecture and solution information were provided. Gemini models had higher scores in comparison to other models, which represents a better capability of information extraction. 'LlaVAModel', at the same time, gained the highest score in the setting without solutions, which indicates the ability of extracting knowledge from massive textual data.

6.2.5. METEOR

TODO

6.2.6. ROUGE

The highest ROUGE scores were observed in setting 3, where both lectures and solutions were available, highlighting the importance of comprehensive input for producing informative responses. Setting 1 showed lower ROUGE scores across all models, reflecting limited informativeness when models were provided with minimal context. 'LlaVAModel' anf 'GPT-4' had highest ROUGE-scores on settings without solutions. Outputs of 'GPT-4' without knowledge of the correct answer had higher ROUGE score than 'GPT-4o-mini' given solutions in the input.

6.2.7. Cosine similarity

Semantic alignment with the reference is assessed by the similarity metric. Gemini and its variants outperformed other models, demonstrating their great capacity to capture the semantic structure of enriched input. Similarity scores were highest in settings 3 and 4. LLaVAModel1.5-7b, on the other hand, received the lowest score in setting 1, indicating that it mostly depends on lecture or solution data to generate outputs that are semantically accurate. This result emphasises how more data affects the resulting content's semantic integrity. 'GPT-4o-mini' and 'LLaVAModel' performed better than other models in environments lacking solution information, indicating that they can produce reliable results even with limited contextual guidance.

6.2.8. Summary

Overall, the findings demonstrate how important enriched contextual data, such as lectures and solutions, in enhancing model performance across all metrics. The GPT-4o-mini model outperformed GPT-4 in multiple instances, indicating the efficacy of smaller, more targeted designs, even if GPT-4 continuously received excellent accuracy and informativeness scores. While LLaVAModel1.5-7b failed without extra input, highlighting its dependence on extensive contextual knowledge to function well, the Gemini family of models demonstrated remarkable robustness and semantic

alignment, particularly in enriched situations. Lecture information doesn't really affect models' performance, it can even decrease the ratio of correct answers.

6.3. Zero-shot Benchmarking on Validation

6.3.1. Local Models from HuggingFace

Pre-experiments We were not successful with processing with small non-chat models. For the given text + image, the models would return short non-meaningful answers, like "Yes" to a "X or Y" question, or "no idea" to a completely defined task with all relevant context included. These were multimodal models below 1B on Huggingface, not tuned to be used in a chat scenario (and prepending a "You are a helpful assistant..." to the beginning of the prompt was not enough to fix this). We are not sure how to use those models for our purposes as of now.

LLaVa The exact model used is: llava-hf/llava-1.5-7b-hf (and there is a larger sister 13b model). It is based on LLaMA and (?) Vicuna fine-tuned on GPT-generated multimodal instruction-following data. It expects inputs as conversation, allowing the preprocessor to convert textual input into format "USER: request ASSISTANT: " where the assistant has to fill in the rest. We encountered the following issues while applying this model version on ScienceQA data:

- although prompted like larger models, this model returned answer strings and not choice index; this may additionally raise an issue of the returned answer not being entirely part of the choice list;
- although prompted to generate JSON, the model would sometimes generate a different string, e.g. simply repeating part of the task. Unexpected format happened in ?? % of all answers. We disregarded these answers, considering them all wrong, which results in a bit more pessimistic estimation of the model's performance. The metrics on the correctly formatted answers alone are also given.
- the model currently does not accept inputs without an image (text-only); according to a huggingface discussion[1], the authors claim to have accidentally removed this possibility in a recent update. We worked around this by producing placeholder empty 10x10 white images and passing them as part of the input when the image in the dataset was missing. We also appended the suffix "Ignore image" to the end of the prompt, because the model would otherwise try to hallucinate a relation between the text and the image (e.g. "Why is the sky blue" + placeholder image results in "The sky is blue on the picture because ...");
- the model does not really seem to support batch processing; although officially the technical possibility exists, it has been reported online [2] that the model would only take into account the first image in the batch, which makes the processing useless. We did not risk to attempt the supported or custom batching and processed the filtered validation dataset 1 datapoint by 1, which took 10+ hours on GPU for inference.

Model name	llava-1.5-7b-hf
Size	7B
Published	09.2023
Type	SLM
Chat model	yes
Parent models	LLaMA, Vicuna
Data	GPT-generated multimodal instruction-following da

Table 2. Model Comparison: Benchmarked Models

There is another available model LLaVANext, which recently started to support batching, and should have better performance.

Note: I later (after running the model) understood that the original model is this one: 'liuhaotian/llava-v1.5-7b'. The results may thus be different from what would be expected from the model's paper.

Qwen models Current experiment

Pixtral

7. Knowledge Distillation & Soft Prompting

8. Error Analysis

9. Conclusion

[1] https://huggingface.co/llava-hf/llava-1.5-7b-hf/discussions/38

[2] https://huggingface.co/llava-hf/llava-1.5-13b-hf/discussions/10