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 pre-trained 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

Recent developments in foundation models have shown remarkable performance on challenges that call for contextual awareness and scientific reasoning. However, these models are often resource-intensive, which limitates their scalability and accessibility for broader applications. Through the transfer of reasoning capacities from large models to smaller, more efficient ones, this study focuses on the possibility of knowledge distillation to address these issues. By applying this approach to the Science QA datase, we want to close the gap between performance and usefulness by enabling lightweight models to handle challenging reasoning tasks with high accuracy and robustness. **TODO**

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

[*]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.

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 ScienceQA dataset is a benchmark for multimodal reasoning in science, consisting of over 21,000 questions across topics like natural science, social science and language science. Each question includes textual prompts, optionally visual aids (e.g., diagrams, charts), answer options, detailed explanations and lecture context. The dataset spans various difficulty levels, enabling evaluation of models on both basic and advanced scientific reasoning, reaching from elementary to high school level questions. ScienceQA supports tasks such as multimodal comprehension, answer explanation generation, and educational AI development. Its high-quality annotations and diverse content make it an ideal benchmark for assessing models in science education and reasoning. In our experiments, we leveraged the dataset to assess knowledge distillation under diverse scientific scenarios, focusing on the ability to distill the reasoning capabilities. Irregularities: For some datapoints, the image data is missing, while the answer can be deduced from text alone. Apart from the image field, missing values can occur in lecture and solution fields. As about 50% 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.

6. Metrics

We evaluate model's performance with question answering accuracy domain-wise in order to have a fair comparison with leaderboards. The reasoning steps can be evaluated with semantic similarity metrics (e.g. adopted from machine translation) such as BLEU, METEOR, ROUGE, and cosine similarity.

6.1. Multiple choice Evaluation

Owing to the simplicity of the test format, only accuracy score is computed, following original evaluation strategy by (**?**). It was determined that the computation of the overall score should assign the highest weight (**50%**) to the ability to provide correct answers (accuracy score), as this represents the most critical aspect of performance evaluation.

6.2. Answer Reasoning Evaluation

Due to the pecularity of scientific texts and approaches to the evaluation of automatically generated texts, the following evaluation approaches were chosen. Our validation process is consisted of the metrics introduced in the original paper (**?**) with a few additional metrics and techniques that we consider of high importance in regards to the scientific textual data.

BLEU BLEU-score ((**?**)) can measure how closely the model's generated explanation aligns with the human-authored explanation. BLEU-1 measures if the model uses the right scientific terms or key words (e.g., "photosynthesis," "temperature"). However, it ignores word order and logical progression, so it can't evaluate reasoning or explanation quality. That is why BLEU-4 score is also computed

in order to capture both vocabulary and the arrangement of words into meaningful phrases, evaluating fluency and coherence. Scientific explanations can often convey the same idea with different wording, making semantic similarity hard to capture. BLEU might penalize valid but rephrased or concise answers and it does not account for deeper meaning or logical correctness, which are crucial in reasoning tasks. Since this metric can give valuable insights on the generated outputs, we do not consider these scores as the most important and both BLEU-1 and BLEU-4 scores have **5%** impact.

ROUGE ROUGE-L score, following evaluation strategy in (**?**), helps analyze the content recall and sequence preservation of model-generated explanations. They highlight whether a model captures key scientific concepts and produces explanations that align with reference reasoning. However, ROUGE should be used in conjunction with other metrics to fully evaluate the logical and factual quality of generated reasoning. ROUGE-L will highlight the mismatch in logical flow, even though some content overlaps in reference and candidate sentences (6.2). It lead to the conclusion that this metric would not drastically affect the overall score and contribute (**5%**).

- **Reference:** Plants absorb sunlight to produce energy.
- **Candidate:** Sunlight is absorbed by plants to create energy.

METEOR METEOR ((**?**)) evaluates text similarity based on precision, recall, and alignment of semantically meaningful components like synonyms and stems, making it well-suited for capturing the nuanced language used in scientific reasoning. By incorporating a weighted F-score and penalties for excessive mismatches, METEOR provides a balanced evaluation of fluency and accuracy, crucial for assessing LLM-generated answers on the Science QA Dataset. Due to these balanced text evaluation, this metric has **15%** impact on the overall score.

Cosine similarity with Sentence Transformers Sentence Transformer models are optimized for sentence-level embeddings, which capture the overall contextual similarity of answers more effectively, making them suitable for general-purpose evaluation tasks like scientific reasoning. Unlike SciBERT, which is tailored to scientific text and could have been used as sentence embedder, Sentence Transformers are pre-trained on diverse datasets, enabling them to generalize better to variations in phrasing and style present in the Science QA Dataset. Their ability to handle broader linguistic nuances ensures that the evaluation accounts for both semantic correctness and the logical flow of responses, which is essential when assessing reasoning across a wide range of scientific topics. That is the reason for using Sentence Transformer model for measurements of cosine similarity between the lecture material and annotated explanations and models' outputs with impact weight of **10%**.

Overall score In order to obtain a numeric description of the model's performance, we set the follwoing metric that includes previously mentioned metrics with the given importance weights:

$$\begin{aligned}\text{Overall Score} = {} & 0.6 \cdot \text{Accuracy} + 0.05 \cdot \text{BLEU-1} \\ & + 0.5 \cdot \text{BLEU-4} + 0.05 \cdot \text{ROUGE} \\ & + 0.15 \cdot \text{METEOR} \\ & + 0.1 \cdot \text{Cosine Similarity}\end{aligned}$$

We prioritize the ability to generate correct answers as the most critical aspect, with reasoning ranked as the second-highest priority in our overall evaluation metric.

7. Experiments

7.1. 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:

1. OpenAI models: `GPT-4`, `GPT-4o-mini`
2. MistralAI model: `Pixtral-12b-2409`
3. Google models: `Gemini-1.5-flash`, `Gemini-1.5-flash-8B`
4. Llava model: `llava-hf/llava-1.5-7b-hf`

7.2. Results

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` and `Gemini-1.5-flash-8B` consistently received highest ratings, especially in scenarios in which lecture and/or solution information was included. The Gemini family of models and `Pixtral-12b-2409` model 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.

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 'Llave-1.5-flash-8B' 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.

BLEU-4 Gemini models had higher scores in comparison to other models, which represents a better capability of information extraction. `GPT-4o-mini`, at the same time, gained the highest score in the setting without solutions, which indicates the ability of extracting knowledge from massive textual data.

METEOR METEOR scores reveal how well the models capture fluency, grammar, and word-level alignment with the references. Similar to cosine similarity, Gemini-1.5-flash leads with the highest METEOR scores. Notably, `Pixtral-12b-2409` achieves relatively high METEOR scores in settings 3 and 4, despite moderate cosine similarity, highlighting its ability to produce fluent outputs. When concrete tasks without any additional helpful material are given to the models as an input, `GPT-4o-mini` model shows a capability of generating concise, fluent answers with correct relevant terminology and explanations.

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.

Cosine similarity Cosine similarity highlights the semantic alignment between the model outputs and the correct answers. Gemini-1.5-flash consistently achieves the highest similarity scores, particularly in settings 3 and 4, indicating strong alignment with the reference answers. `GPT-4` variants maintain steady scores across all settings, demonstrating robustness, while `Pixtral-12b-2409` and `LLaVAModel1.5-7b` show more variability, suggesting sensitivity to specific settings. Overall, cosine similarity shows that Gemini-1.5-flash excels in semantic understanding across the dataset.

Overall score The overall results indicate that the 'GPT' and 'Gemini' model families demonstrate exceptional ability in extracting accurate answers. Notably, the 'Gemini' models exhibited superior performance on datasets lacking relevant lecture information. The second-best performance on such "pure" datasets was achieved by `GPT-4`, with only a slight difference in overall score.

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.

7.3. Zero-shot Benchmarking on Validation

7.3.1. 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, and is 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, as we discovered later, the model had actually been trained on ScienceQA):

- 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.

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

[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

Pixtral We benchmarked `mistralai/Pixtral-12B-2409` on BWCluster using `vllm` and 4 GPUs. In this setup, the generations may continue beyond a stop token, degenerating into complete nonsense (code, crazy text, gibberish). A postprocessing script was used to cut answers of the model when the first stop signal occurs (closing ¡s¿ tag, or [STOP] etc).

A bad degeneration case was when the model tried to hallucinate a similar example, producing a new [INST] item with a well-formatted "question", and then answered it. It could happen multiple times within one generation. We explicitly checked that such questions are not part of the validation set that we were using for benchmarking and thus not artefacts of bad processing from our side (one could suggest that multiple questions were passed as input instead of one, yet it was not the case).

We noticed similar behaviour with degenerations in our `qwen` experiments, also run in a similar `vllm` setup.

Qwen models We are currently benchmarking `Qwen/Qwen2-VL-2B-Instruct`, and are planning to use this model as a basis for our soft prompting experiments. Qwen models are reportedly good for reasoning; in these models, multimodality starts at 2B parameters.

8. Knowledge Distillation & Soft Prompting

TBD: as mentioned on the slides, we will conduct the described experiments and thoroughly report the results later.

Soft Prompting Goal: Investigate how well knowledge distillation works compared to prompt tuning for science reasoning tasks and to zero-shot inference

We freeze a LLM and learn prefix to guide the model Prompt tuning increases/decreases the performance by xThis is further compared to Knowledge Distillation from outputs of Teacher model, which performs insert here in comparison to prompt tuning

Soft Prompting & Knowledge Distillation Goal: Investigate how well knowledge distillation works compared to prompt tuning for science reasoning tasks.

This is further compared to Knowledge Distillation from outputs of Teacher model, which performs insert here in comparison to fine tuning We try different prompting techniques to obtain useful outputs for learning of the Student Prompting technique X results in the best performance

Soft Prompting & Knowledge Distillation Goal: Investigate if data distilled from Teacher can be (more) useful as compared to using just the training data for science reason-

ing tasks.

Agentic Setup & Knowledge Distillation Goal: Investigate if data distilled from Teacher + retrieved sources (+ explanation of retrieved data by Teacher) allows us to learn a better Student model. Compare to: Student + RAG capabilities

9. Error Analysis

10. Conclusion

References