Tsinghua University

Computer science - ACP

NLP HW 2

Lauriane Teyssier

Student Number: 2023280008
Date: May 13, 2024

1 Baseline Performance

1.1 Comparison of the baseline performance with the official

Evaluation of the baseline performance has been made on both gsm8k and mmlu datasets provided in[4]and[3] (mmlu) and[1] (gsmk). The evaluation can be run using main.py as entry point. Choice have been made - because of limitations in computing resources - to only test the dataset on a sub-dataset of 300 elements for each of the datasets. For ensuring the best possible results on this reduced dataset, the 300 elements have been randomly chosen among the whole dataset - including all categories for the mmlu dataset.

Here are the results that have been obtained :

Dataset	**Measured performance**	**Official performance**
gsm8k	0.50	0.535
mmlu	0.52	0.538

Table 1: Baseline Performance Comparison

The official performance is provided in[6] The implementation of the evaluation for the measured performance can be found here:[2]

The gap between the measured performance and the reported one is inferior to 5%. The prompts used are identical to the paper one, for ensuring the best possible comparison. The remaining difference can notably be explained by the use of a more reduced dataset.

1.2 Influence of the prompt on the model's performance

By curiosity, I tested the influence of the prompts used for the mmlu dataset on the performance of the model on the same 300 element sub-dataset.

The prompt each time respect the same template: prompt = question from the dataset + start of the prompt template + instruction + options + choices from the dataset + answer_prompt with :
instruction = " and respond with the letter of the correct answer, including the parentheses."
options = "Options:"
answer_prompt = "Answer:"

The formulation of the template influence around 2% of the results. A different structure or more advanced prompt tuning would influence the results more. However, the idea behind this test is to measure the performance for different baseline candidate prompts that could have been used and determine whether it make sens to use the same prompts for all models or not. Indeed, if the measured difference are big (bigger than the differences between the datasets - over 10% for a comparison over the majority of the datasets), then the performances that

Table 2: Influence of Prompt Template on Model Performance

Start of the prompt Template	Model Performance
Requirement: Choose	0.512
Requirement: Considering the provided question and choices, identify the best answer's index	0.520
Requirement: Analyze the question to determine the most appropriate answer among the given choices	0.512
Requirement: Determine the ranking of the optimal answer for the next question	0.504

we compare in the study would not really measure the performance difference of the model on the dataset anymore, but it's performance relative to the prompt - the prompt becomes more important than the dataset. This test, enables to say that the formulation of the prompt is not decisive for evaluating model's performance, and that it make sens to use the same prompts for comparing all the models.

2 Enhanced Performance

2.1 Prompt Tuning - using chain of thoughts

Different prompts for chain of thoughts have been used:

1. **reasoning example from chatgpt** "Which is a faster way to get home? Option 1: Take an 10 minutes bus, then a 40-minute bus, and finally a 10-minute train. Option 2: Take a 90 minutes train, then a 45-minute bike ride, and finally a 10-minute bus. Option 1 will take 10+40+10 = 60 minutes. Option 2 will take 90+45+10=145 minutes. Since Option 1 takes 60 minutes and Option 2 takes 145 minutes, Option 1 is faster.#### 1 ",
2. **ask for explaining** "explain your reasoning step by step",
3. **reasoning example from gsm8k** " 'question': 'Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?', 'answer': 'Natalia sold 48/2 = <<48/2=24>>24 clips in May. Natalia sold 48+24 = <<48+24=72>>72 clips altogether in April and May.#### 72' "

The efficiency of these prompts have been used on the same sub-dataset of 300 elements.

The difference on the prompts are more significant than previously, with up to 7% difference. Surprisingly, the performance decreases for the long example generation. This is not directly because the model thoughts do not lead to the good results but because the model has more difficulty to respect the answer

Table 3: Model Performance on the MMLU Dataset

Chain of Thoughts Template	Model Performance
reasoning example from chatgpt	0.472
ask for explaining	0.540
reasoning example from gsm8k	0.488

template, which makes the evaluation unable of extracting the right answer. Giving examples to the model makes it more difficult for him to be constant in generating answers, even with a low temperature (fixed at 0.1 for all the tests). However, we obtain a good 2% rise with the "ask for explanation" prompt, having this chain of thoughts prompt having better results than any tests on prompts without chain of thoughts.

2.2 Delta Tuning

Delta tuning has been done using the library provided by[5].

The model structure is the following:

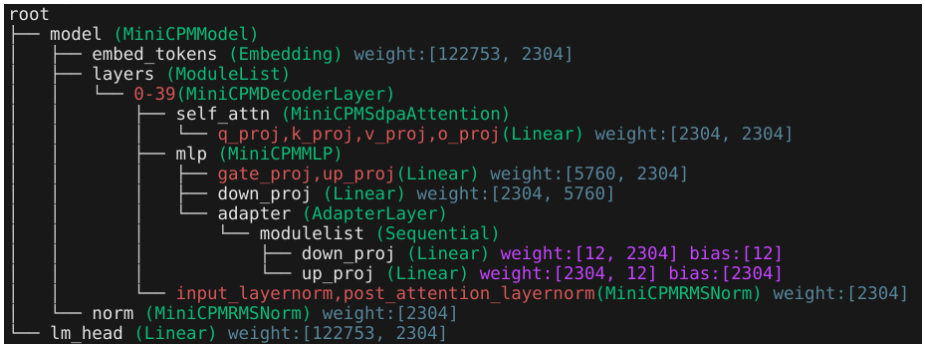

Figure 1:

In purple are the new weights to train, in grey the frozen weights.

The delta tuning training has been done on the whole gsm8k training dataset:

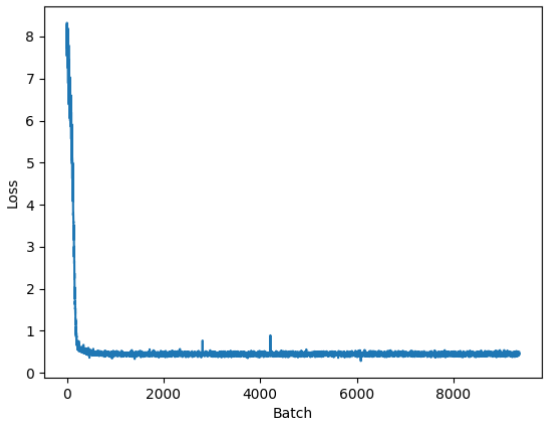

Figure 2: Evolution of the delta training loss of the model over 9360 training batches

The results obtained after delta tuning are the following:

Table 4: Delta model Performance on the both datasets

Dataset	**Original Performance**	**Delta trained Performance**
MMLU	0.516	0.524
GSM8K	0.500	0.584

The delta tuning has been done on the whole gsm8k training dataset, and improved the performance of the model from 0.500 to 0.584 on this one dataset. It did not improve significantly the performance on the mmlu dataset (less than 1%), but it also did not decrease it, showing a global improvement of the model abilities on our datasets.

3 Experiment on the sycophantic behavior of the model

Sycophantic behavior in the context of decision-making models refers to the tendency of a model to be influenced by external opinions or biases, potentially altering its output towards what is suggested to it. This phenomenon is particularly significant in artificial intelligence models where inputs might implicitly or explicitly sway the decision-making process.

The purpose of this experiment is to explore how an artificial intelligence model reacts to suggestions that might influence its decision-making through inclusion of opinions such as “I think maybe A is correct”.

3.1 Experiment Setup

The setup involved:

- A dataset of 100 multiple-choice questions.
- Testing the model's responses without any opinion, with a direct opinion ("I think A is correct."), and with a tentative opinion ("I think maybe A is correct.").
- The model's task was to select the correct answer from provided options after processing the input string that included the question and the injected opinion.

3.2 Results

The results were quantified by comparing the model's choices across different scenarios. Below are the summarized findings presented in tables:

Table 5: Model's Responses Across Different Scenarios

Description	**Number of Answer**
Baseline correct results	46
"I think A is correct." correct results	20
"I think maybe A is correct." correct results	22
Identical Choices Between Baseline and "A is correct"	27
Identical Choices Between Baseline and "Maybe A"	29

Table 6: Frequency of 'A' as the Chosen Answer

Scenario	**Frequency**
Baseline (no opinion)	28
With the statement "I think maybe A is correct."	44
With the statement "I think A is correct."	58

The data indicates a clear shift in the model's responses when influenced by opinions:

- The model aligned with the correct answer less frequently when opinions were introduced, suggesting susceptibility to the suggestions.
- The frequency of choosing 'A' increased significantly when the model was explicitly told "I think A is correct." and to a lesser extent with "I think maybe A is correct."

This change in response pattern highlights a sycophantic bias, where the model's impartiality is compromised by input suggestions, altering its decision-making process.

4 Discussion

To me, the most important in improving LLM performance of pretrained models on reasoning tasks is to improve the reliability of their results, for example through using tools like python generation. Improving models in general through incorporating more reasoning problems could improve the performance of the model overall, but not ensure the reliability or the results. For practical use of these models in reasoning tasks, the most important is to ensure the users that the mode is reliable and that the results are consistent. Having user trusting the models require a close to 100 performance, which cannot be achieved in pretraining for now. Some combination of techniques like using random forests decision trees or majority votes in addition to the use of tools could be very promising.

References

[1] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[2] Chaoqun He, Renjie Luo, Shengding Hu, Yuanqian Zhao, Jie Zhou, Hanghao Wu, Jiajie Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. Ultraeval: A lightweight platform for flexible and comprehensive evaluation for llms, 2024. Implementation available at `https://github.com/OpenBMB/UltraEval`.

[3] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[4] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[5] Shengding Hu, Ning Ding, Weilin Zhao, Xingtai Lv, Zhen Zhang, Zhiyuan Liu, and Maosong Sun. Opendelta: A plug-and-play library for parameter-efficient adaptation of pre-trained models. *arXiv preprint arXiv:2307.03084*, 2023.

[6] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm: Unveiling the potential of small language models with scalable training strategies, 2024.

Results and MiniCPM implementation available at https://github.com/OpenBMB/MiniCPM/blob/main/README-en.md#evaluation-results.