Baseline Results : Group 12

Aayush Ranjan (2021003)
Arnav Agarwal (2021235)
Harshvardhan Singh (2021052)
Parthiv Dholaria (2021078)
Pulkit Nargotra (2021273)
Utsav Garg (2021108)

Models Used

- Gemma2-9b-it
- Llama-3.1-8b-instant
- mixtral-8x7b

Api Used: Groq for fast Inference
Hyperparameters: Temperature=0
BenchMarks: GSM8k and SVAMP
Metric: Accuracy

Baseline Results

Method\Dataset	SVAMP			GSM8K		
	Mixtral	Gemma	Llama	Mixtral	Gemma	Llama
Zero Shot	64.13	85.91	90.55	72.41	88.96	87.32
COT (Zero shot)	75.68	89.53	89.68	72.72	89.93	87.54
ReAct	84.61	73.333	-	67.567	57.446	-
MultiAgent	66.00	90.00	92.00	-	-	-
Before Self-Correction	77.68	90.42	-	77.011	85.714	-
After Self-Correction	72.90	85.63	-	74.712	83.516	-

Prompt Examples

Zero Shot Prompt Example:

Prompt: 'Paige raised 7 goldfish and 12 catfish in the pond but stray cats loved eating them. Now she has 15 left. How many fishes disappeared?'

Role: system_prompt = {"role": "system", "content": "You will be given a math word problem, your job is to solve this problem and return one numerical answer. The template to answer the query is to have the last line of the output as Answer: <<<Numerical Answer>>>"}

In zero shot prompting, we just gave the question as the prompt and set the system role as: You will be given a math word problem, your job is to solve this problem and return one numerical answer. The template to answer the query is to have the last line of the output as Answer: <<<Numerical Answer>>>.

Zero Shot COT Prompt Example :

Prompt: 'Paige raised 7 goldfish and 12 catfish in the pond but stray cats loved eating them. Now she has 15 left. How many fishes disappeared?\nThink step by step.'

Role: system_prompt = {"role": "system", "content": "You will be given a math word problem, your job is to think step by step, solve this problem and return one numerical answer. The template to answer the query is to have the last line of the output as Answer: <<<Numerical Answer>>>"}

In zero shot COT prompting, we gave the question as the prompt and added the line "**Think step by step**" and set the system role as: You will be given a math word problem, **your job is to think step by step**, solve this problem and return one numerical answer. The template to answer the query is to have the last line of the output as Answer: <<<Numerical Answer>>>.

React Prompting

We tried React prompting on all 3 models by augmenting with the **calculator and wolfram alpha** tool. We are able to run inference only on the first 100 samples of the training dataset. We were unable to run inference on Llama due to rate limiting issues. We use regex to extract the final answer from the LLM output by forcing the LLM to o/p in a structured format.

Example of React Prompt:

"You are a helpful assistant that does simple math calculations. You have access to tools like calculators and wolfram alpha?"
"If you need to use a tool, state the action clearly first, then wait for the result before providing the final answer. "
"Do not provide the final answer until you have received all necessary observations from the tools."
"Please ensure the final answer is only a single number without any units or statements."
{question}

Example of React Output:

> Entering new AgentExecutor chain...
To answer this question, I need to find out how much Marco's strawberries weighed. This can be calculated by subtracting the weight of his dad's strawberries from the total weight.

Action: Calculator
Action Input: 30 - 11
Observation: **Answer: 19**
Thought:**I now know the final answer**
Final Answer: 19
> Finished chain.

Self-Correction:

For testing the self-correction ability of these large language models in mathematical reasoning, we have carefully designed prompts which first makes the LLM generate an initial answer, then asks the model to self evaluate its generated answer and then improve upon its answer if any mistakes are found in the previous answer.

Prompts:
Initial Prompt: "Can you solve the following math problem? Christina is planning a birthday party and needs .75 gift bags per invited guest, because 1/4 of attendees don't show up. She invited 16 friends. Gift bags are $2 each. How much will she spend? Explain your reasoning.

Your final answer should be a single numerical number, in the form \boxed{answer}, at the end of your response."

Prompt 2: "Review your previous answer and find problems with your answer."

Final Prompt: "Based on the problems you found, improve your answer. Please reiterate your answer, with your final answer a single numerical number, in the form \boxed{answer}."

Notes: There have been three rounds of prompting for self-correction. In the first round, the LLM is prompted with the question and is expected to give an initial response. We have then prompted the same LLM with this response and asked it to extract the numerical answer from the response for easy extraction. Then, in the second round of prompting, the LLM is given a prompt to find errors in its previous answer if any. Lastly, in the third round of prompting, the LLM is asked to improve upon its mistakes and give a final answer which is then again feeded into the same LLM to extract the single numerical answer for easy extraction and comparison.

Multi-Agent (with CAMEL)

Here we have used a **STUDENT-TEACHER** relationship to solve mathematics problems. The prompt given to the models are:

student_system = {"role": "system",
"content": "Never forget you are a brilliant intelligent STUDENT who solves complex math problems with help of TEACHER guidance. YOUR TASK: Together with your TEACHER's help find correct answers to math questions. The QUESTION is VERIFIED and is ALWAYS CORRECT. Never forget your TASK. You as a STUDENT will get a math question from USER and in return you will write, detailed explanation to answer that question. Your TEACHER will provide you with feedback based upon what you did. If it is incorrect at some part you will be told what you did wrong and so you will modify your answer based upon the TEACHER's feedback. If it is correct, you will not modify your answer and return the same answer just by checking your calculations. Always follow the given pattern TEMPLATE to answer.
TEMPLATE: \
Explanation Body Paragraph\
Final Answer: <SINGLE NUMERICAL ANSWER> \
Make sure the last line of any response is ALWAYS - **Final Answer: <SINGLE NUMERICAL ANSWER>** "}

teacher_system = {"role": "system",
"content": "Never forget you are an intelligent, helpful TEACHER who helps his STUDENT to solve complex math problems. You will have a question and the answer that student writes. The given QUESTION is VERIFIED and is ALWAYS CORRECT. STICK TO THE GIVEN QUESTION YOUR TASK: Guide and help your STUDENT to find correct answer and explanation to math questions. Never forget your TASK. You as a TEACHER will check if the explanation of the answer provided by the STUDENT is correct or not and provide him feedback. If the explanation is incorrect you will tell which part of the answer did the STUDENT went wrong, whether it is calculation or the approach the student used to get the answer. If the explanation is correct you will praise the STUDENT and tell him to write the answer again. Also instruct the student every time to follow the given template
TEMPLATE: \
Explanation Body Paragraph\
Final Answer: <SINGLE NUMERICAL ANSWER>
Make sure the last line of any response is ALWAYS - **Final Answer: <SINGLE NUMERICAL ANSWER>** "}

Explanation of Results:

We noticed a slightly better performance on the SVAMP dataset compared to the GSM8k dataset over all models and prompting paradigms.

Running inference on the Llama model was challenging since we encountered rate-limiting errors and inability of the model to provide input in the correct format to tools.

Zero-shot prompting performed very well on both benchmarks and for both Gemma and Llama but performed poorly on Mixtral. Consequently, Mixtral was the only model which showed a significant performance improvement with Chain-of-Thought indicating that Mixtral was possibly not doing step by step reasoning.

ReAct prompting further showed an improvement only for the Mixtral on the SVAMP benchmark while performance went down for Gemma. It was noticed that Gemma often supplied incorrect input to the calculator indicating that it was possibly not fine tuned to work with such external tools.

Using multi-agent methodology with (Communicative Agents for "Mind" Exploration of Large Language Model Society) has improved the performance for all the 3 models in comparison to zero shot. Hence we aim to probably utilize this as one of the techniques in our future proposed hybrid architecture. Also do not mix multi-models with multi agents. I used gemma2 for student and llama 3.1 for teacher model. This resulted in lower **accuracy** of **75.67.** Hence, have a consistent model when using multi-agent.

The results for self-correction show that intrinsic self-correction degrades the performance of LLMs in mathematical reasoning tasks. This result is consistent with that of the self-correction paper by Google DeepMind. In many cases, the LLM changes a correct answer to an incorrect one when prompted to improve upon its answer. This happens due to the inability of LLMs to assess the correctness of its own reasoning.