LLM A2

Part A: Comparisons and Trade-offs

Accuracy:

	Zero-Shot	Chain-of-Thought	ReAct
Gemma	30%	31%	18%
Phi	26%	30%	28%
LlaMa	33%	33%	29%

Time/Prompt:

	Zero-Shot	Chain-of-Thought	ReAct
Gemma	0.45s	0.53s	6.23s
Phi	28.67s	27.86s	29.13s
LlaMa	46.06s	34.19s	68.76

Model Size, Inference Time & Accuracy: The results indicate that larger model size often comes with more informative outputs and rich reasoning but we need to pay with increased memory footprint and inference time. But they also indicate that performance does not always degrade with model size as in case of Gemma which indicates the potency of distillation to transfer the capabilities of a larger model to a smaller model.

Prompt Type, Inference Time & Accuracy: As expected ReAct prompting takes the longest inference time in general due to the iterative nature and multi-step reasoning nature of ReAct prompting along with the overhead of querying an external tool. Times of CoT and Zero-shot are comparable and even less for Llama, this could be because while CoT comes with additional overhead of long step by step reasoning it also reduces cognitive load on the model. When using CoT, the model processes each part of the problem sequentially. This can reduce the cognitive load in each step since it's tackling smaller chunks of the task at a time. This, nonetheless, is a surprising result.

Output Quality & Model Size: The relationship between informativeness and model size is generally positive, as larger models tend to produce higher-quality outputs due to their greater capacity to capture complex patterns, nuanced language, and intricate reasoning. This enhanced ability results from having more parameters that allow larger models to learn richer representations from training data. However, beyond a certain point, the gains in output quality may diminish, as other factors like training data quality, fine-tuning, and architectural optimizations also play a crucial role in determining the overall performance. While Phi and Llama do have better output quality, the similar performance to Gemma indicates that informativeness does not always mean accuracy.

Output Quality &

Part B: Technical Report

Model Size:

Google-gemma has 2 billion parameters and it is the smallest model. Microsoft-phi 3.5 mini-instruct has 3.8 billion parameters. The llama-3.1 is the largest model with 8 billion parameters.

All three models show almost an equal performance irrespective of model size with llama only showing a slight improvement. The poor performance of all three models is indicative of the difficulty LLMs have on mathematical tasks in general. Another reason that gemma performs as well as the much larger llama could be due to the fact that gemma has been distilled from a 27 billion parameter teacher Gemma 2 model. Phi- with 3.8 billion parameters shows degraded performance since it has not been distilled from a larger model and has not been trained on any special mathematical data.

This shows that increase in model size can in general lead to improved performance but distillation can be a potent strategy to reduce inference time while keeping performance relatively high.

Inference Speed:

As expected the 2 billion parameter gemma model shows significantly lesser inference time due to small size while phi and llama have much larger inference time.

Gemma has such low inference time due to specific focus on certain inference efficiency improvements:

1) **Grouped-Query Attention (GQA):** Gemma 2 uses the Grouped-Query Attention mechanism, which is shown to be faster at inference time compared to traditional Multi-Head Attention (MHA).
2) **Local Sliding Window and Global Attention**: The model employs a combination of local sliding window attention and global attention, which optimizes how attention is distributed across tokens, improving efficiency during inference.
3) **Logit Soft-Capping:** By capping logits within the self-attention layers and the final layer using the function logits ← soft_cap * tanh (logits / soft_cap), the model limits the range of logit values, which reduces computational overhead during inference
4) **Distillation from Larger Models:** The Gemma 2 2B and 9B models are trained using knowledge distillation from larger models, which helps them learn efficiently, potentially reducing inference time by allowing the smaller models to achieve comparable performance without requiring complex calculations

The reason Phi takes very high inference time is due to lack of distillation or pruning of its parameter space. There has been no architectural focus on improvement of inference time.

Llama particularly focuses on improving inference time with the following key improvements which is why the inference time of the model is low despite the larger size of the model:

1) Pipeline Parallelism: This strategy allows multiple micro-batches to be processed concurrently, significantly enhancing throughput without compromising latency
2) FP8 Quantisation: The Llama 3.1 model employs FP8 quantization for low-precision inference. By quantizing most matrix multiplications in the feedforward network layers, which account for around 50% of inference compute time, the model achieves considerable speedups.

Training Data:

The Gemma 2 training data consists of 2 trillion tokens for the 2 billion model. The data comes from a variety of sources, including web documents, code, and scientific articles, and is primarily in English. his diverse dataset likely contributes to some ability to handle mathematical tasks, particularly given the inclusion of scientific articles and code data, which often contain mathematical reasoning and expressions.However, the training data isn't explicitly optimized for mathematical accuracy or problem-solving, which might limit the model's performance on more complex or nuanced mathematical tasks. Since the 2 billion model is distilled from the 27 billion model it performs very well for its size since the larger teacher model is able to capture information from the training data very well.

The Phi-3 mini-instruct 3.8 billion parameter model has undergone rigorous training on a heavily filtered dataset that emphasizes logical reasoning and language understanding. This training involved a mix of publicly available web data and synthetic data. The Phi-3.8 billion model's training, however, has limitations when compared to larger models. It seems to lack the sheer volume of high-quality mathematical data necessary to build robust mathematical reasoning capabilities at the same level as larger models like Llama-3.1 . This might limit its performance on more complex mathematical tasks despite its efficiency and optimized data approach.

The Llama-3.1 8-billion model is trained on a larger dataset with improved quality and diversity compared to previous versions, including Llama 2. This model benefits from a significant proportion of mathematical and reasoning tokens in its training data—about 25% of the total data mix. This extensive training on mathematical and reasoning data helps it achieve better accuracy on mathematical tasks, as it enhances the model's ability to handle complex mathematical reasoning problems.

Type of Prompt:

We notice Gemma shows similar performance across zero-shot and chain-of-thought prompting which is because these prompting techniques don’t differ significantly as CoT just adds the line ‘Think step by step’. However the performance degrades with ReAct prompting possibly because the model is unable to make use of the calculator tool. Moreover it might be because unlike the other two models Gemma has not been fine tuned for language reasoning and is unable to make sense of structured reasoning prompts.

Phi model shows an improvement from zero-shot to chain-of-thought possibly because of the fine tuning on language reasoning tasks. This performance boost indicates that the Phi model has a stronger ability to handle tasks that need more structured thinking and logic compared to straightforward problem-solving. Phi also shows similar performance with ReAct prompting because it is able to use reasoning chains efficiently. Maybe the use of stronger tools like Wolfram Alpha could have further improved the performance of ReAct on phi model.

Llama showed similar performance with zero-shot and chain-of–thought prompting since these prompting techniques don't differ significantly as CoT just adds the line 'Think step by step'. However it was significantly surprising to see llama show a degradation in accuracy with ReAct prompting since it is explicitly fine tuned to use tools like Wolfram Alpha, while also being instruction fine tuned and further trained on language reasoning. The only reasons I could think for this are:

1) While LLaMA 3.1 is fine-tuned on tool usage, the ReAct framework requires the model to not only use tools but also integrate the output from these tools into its reasoning in a multi-step fashion. This can introduce more cognitive complexity compared to simpler tool-usage scenarios.
2) Despite being fine-tuned for tool usage, LLaMA might still have a tendency to default to internal reasoning, especially if it has been trained extensively on purely reasoning tasks. When using ReAct, the model might not be leveraging the tool outputs as effectively as it should, leading to reduced accuracy.

Output Quality:

I noticed that Gemma had poor informativeness with short explanations even on react prompting showing lack of language reasoning and mathematical reasoning capabilities. However, it still had a decent accuracy possibly because it had been distilled from a larger model.This could also be because it could identify the correct option based on pattern recognition or shallow correlations within the prompt, even if it didn't exhibit deep understanding or logical reasoning.

In the zero-shot setting, the Phi model receives the prompt without additional guidance or structured reasoning cues. This means it has to generate an answer without the benefit of breaking down the problem, leading to less informative and sometimes superficial responses. However phi was able to significantly improve its performance on CoT and ReAct paradigms as it was guided by a step by step reasoning process. While this reasoning process was not sound proof it encouraged the model to articulate the thought process.

Llama unlike phi and gemma produced informative outputs with all prompting paradigms showing the benefit of having a larger model size and explicit fine tuning on mathematical tasks. However the accuracy was not much better than gemma or phi which indicates presence of mistakes and errors in its reasoning and thought process.

References:

1) Gemma 2: Improving Open Language Models at a Practical Size: https://arxiv.org/pdf/2408.00118
2) Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone: https://export.arxiv.org/pdf/2404.14219
3) The Llama 3 Herd of Models: https://scontent.fdel27-2.fna.fbcdn.net/v/t39.2365-6/453304228_1160109801904614_7143520450792086005_n.pdf?_nc_cat=108&ccb=1-7&_nc_sid=3c67a6&_nc_ohc=9omwM7QYS0YQ7kNvgFdDCLS&_nc_ht=scontent.fdel27-2.fna&_nc_gid=A3O8KswIjjWTE_hVo2e_3ky&oh=00_AYDnK451XiJc1CRoR6InAeHg5A5MHCl1vLFrMDyUjaufRQ&oe=66F60C47
4) Experiment ReAct prompting with Llama 2 70B-chat :https://www.linkedin.com/pulse/experiment-react-prompting-llama-2-70b-chat-limin-ma
5) LangChain API Docs :https://api.python.langchain.com/en/latest/langchain_api_reference.html#module-langchain.agent