LLM Assignment 2 Report
Harshvardhan Singh, 2021052

Zero shot Prompting

```
print("Total inference time:",end-begin,"seconds")
print("Accuracy score:",100*calculate_accuracy(predictions_zeroshot_gemma,answers)
```

```
Zero shot prompting on Gemma
Total inference time: 215.90406322479248 seconds
Accuracy score: 28.999999999999996 %
```

+ Code + Markdown

```
print("Accuracy score:",100*calculate_accuracy(predictions_zeroshot_phi,answers)
```

```
Zero shot prompting on Phi
Total inference time: 569.4420506954193 seconds
Accuracy score: 34.0 %
```

+ Code + Markdown

```
print("Zero shot prompting on llama")
print("Total inference time:",end-begin,"seconds")
print("Accuracy score:",100*calculate_accuracy(predictions_zeroshot_llama,answers),
```

```
Zero shot prompting on llama
Total inference time: 551.2141418457031 seconds
Accuracy score: 38.0 %
```

In Zero Shot prompting, Meta Llama stands out with a performance of 38%, outperforming the other models and highlighting its strong generalization abilities without requiring additional context or step-by-step reasoning. Its larger size, with 8 billion parameters, likely contributes to its superior handling of mathematical problems. Microsoft Phi follows closely with a score of 34%, demonstrating solid generalization capabilities but not quite matching Meta Llama's effectiveness in tackling unguided tasks. Meanwhile, Google Gemma lags behind with 29%, suggesting that it struggles more with mathematical problems in Zero Shot scenarios, where no extra context or structure is provided.

Chain of Thought Prompting

[21]:

```
print("Chain of thought prompting on Gemma")
print("Total inference time:",end-begin,"seconds")
print("Accuracy score:",100*calculate_accuracy(predictions_cot_gemma,answers),"%")
```

```
Chain of thought prompting on Gemma
Total inference time: 230.00375247001648 seconds
Accuracy score: 30.0 %
```

```
print("Total inference time:",end-begin,"seconds")
print("Accuracy score:",100*calculate_accuracy(predictions_cot_phi,ar
```

```
Chain of thought prompting on Phi
Total inference time: 1211.9865200519562 seconds
Accuracy score: 34.0 %
```

+ Code + Markdown

```
14]:
print("Chain of thought prompting on Llama")
print("Total inference time:",end-begin,"seconds")
print("Accuracy score:",100*calculate_accuracy(predictions_cot_llama,answers),"%")
```

```
Chain of thought prompting on Llama
Total inference time: 1022.6764314174652 seconds
Accuracy score: 32.0 %
```

In Chain of Thought prompting, Microsoft Phi leads with a performance of 34%, showing that it excels at handling tasks that require step-by-step reasoning. This suggests that the model's architecture is well-suited for tasks that benefit from structured, logical prompts, likely due to its training focus on instruction-following tasks. Meta Llama follows closely with a score of 32%, indicating it also benefits from structured reasoning, though not as effectively as Phi. Google Gemma sees a slight improvement with Chain of Thought prompting, scoring 30%, which suggests that it benefits from guided steps in reasoning but still falls behind Phi and Meta Llama in overall performance.

ReAct Prompting

```
print("Total inference time:",end-begin,"seconds")
print("Accuracy score:",100*calculate_accuracy(predictions_react_gemma,answers),"%")
```

```
ReAct prompting on Gemma
Total inference time: 200.5608777999878 seconds
Accuracy score: 23.0 %
```

```
print("Accuracy score:",100*calculate_accuracy(predictions_react_phi,answers),"%")
```

```
ReAct prompting on Phi
Total inference time: 3160.4180772304535 seconds
Accuracy score: 22.0 %
```

+ Code + Markdown

```
22]:
print("ReAct prompting on Llama")
print("Total inference time:",end-begin,"seconds")
print("Accuracy score:",100*calculate_accuracy(predictions_react_llama,answers),"%")

ReAct prompting on Llama
Total inference time: 3437.0630600452423 seconds
Accuracy score: 19.0 %
```

In React Prompting, Google Gemma surprisingly leads with 23%, outperforming Microsoft Phi (22%) and Meta Llama (19%). This result is unexpected, given that React Prompting is more dynamic and requires the models to extract answers from a chain of reasoning or multiple steps. Gemma's better performance may suggest it handles certain reasoning-based tasks more effectively, though overall accuracy remains low across the board. Microsoft Phi follows closely with 22%, indicating that its ability to extract answers or navigate multi-step reasoning is slightly less reliable in this format. Meta Llama, which excelled in Zero Shot and structured reasoning, underperforms here with 19%, showing that its strengths don't translate as well when dynamic, interactive reasoning is required. However, given the complexity of answer extraction in React Prompting, these results may not be as strong indicators of the models' true capabilities compared to the previous scenarios.

In general, we can see as the model size increases, the inference speed decreases and thus the inference time increases. Also, for Gemma, the inference time does not vary much across different prompting techniques. However for Phi and Llama, the inference time increases as the prompt size increases ie. React takes the most time followed by Chain of thought followed by Zero shot prompting.

Google Gemma consistently has the fastest inference times across all three prompting types, making it the most efficient model in terms of computational cost. This efficiency comes from its smaller parameter size (2B), which allows it to process prompts quickly. It's ideal for real-time applications where speed is more important than the highest accuracy.
Phi takes more time than Gemma but less than Llama because it has 3.5B which lies between both of them.
Meta LLama (8B) (even the quantized version) is a slower model because of its huge size. So LLama is the slowest of all models compared.

Github: https://github.com/Harshvardhan21052/LLM-Assignment-2.git

References:
1) https://arxiv.org/pdf/2403.08295
2) https://arxiv.org/pdf/2404.14219
3) https://arxiv.org/pdf/2302.13971