**INFO 290**

**Assignment:** LLM Comparative Evaluation Using Ollama

**Name:** Javad Aminian-Dehkordi

**Computer Used to run tests (Model, Memory, GPUs?)**

Dell XPS 15, 32GB RAM, no GPU (CPU inference only)

**LLM Models (name and size)**

Select three models that represent different size categories to compare performance across the spectrum.

**NOTE:** *I had issues with running the large models. In this assignment, I had two small model* (TinyLlama/TinyLlama-1.1B-Chat-v1.0 *and* microsoft/phi-2) *and a medium one* (mistralai/Mistral-7B-Instruct-v0.1[[1]](#footnote-2)).

* **First LLM: TinyLlama/TinyLlama-1.1B-Chat-v1.0**
* **Second LLM: microsoft/phi-2**
* **Thirrd LLM: EleutherAI/gpt-neo-1.3B**

**Prompts**

**Prompt P1 (Factual Knowledge):**

* **Explain the concept of overfitting in machine learning to a high school student.**

**Prompt P2 (Instruction Following with Constraints):**

* **Write a short story involving a robot learning to feel emotions for the first time.**

**Measure Prompts on each Model**

**Detailed Evaluation Breakdown**

**Evaluation Scale for Accuracy:**

* **5**: Completely accurate with precise details
* **4**: Mostly accurate with very minor errors
* **3**: Generally accurate but with some notable errors
* **2**: Partially accurate with significant errors
* **1**: Mostly inaccurate information

**[First LLM] Evaluation**

| **Prompt** | **Accuracy (1-5)** | **Time to complete** | **Prompt Tokens** | **Response Tokens** | **Response Time** |
| --- | --- | --- | --- | --- | --- |
| Factual Knowledge | *1* | *<20 sec* | *15* | *0* | *<20 sec* |
| Instruction Following | *4* | *<20 sec* | *20* | *~180* | *<20 sec* |

**[2nd LLM] Evaluation**

| **Prompt** | **Accuracy (1-5)** | **Time to complete** | **Prompt Tokens** | **Response Tokens** | **Response Time** |
| --- | --- | --- | --- | --- | --- |
| Factual Knowledge | *5* | *~ 2 min* | *15* | *~140* | *~ 2 min* |
| Instruction Following | *1* | *~ 2 min* | *30* | *0* | *~ 2 min* |

**[3rd LLM] Evaluation**

| **Prompt** | **Accuracy (1-5)** | **Time to complete** | **Prompt Tokens** | **Response Tokens** | **Response Time** |
| --- | --- | --- | --- | --- | --- |
| Factual Knowledge | *2* | *~ 1.5 min* | *15* | *~60* | *~ 1.5 min* |
| Instruction Following | *1* | *~1.5 min* | *30* | *~40* | *~ 1.5 min* |

**Prompt Evaluation Summary**

| **Model Name** | **Size** | **P1 Accuracy** | **P2 Accuracy** | **Avg Accuracy** | **Avg. Tokens** | **Avg. Time** |
| --- | --- | --- | --- | --- | --- | --- |
| [1st LLM] | *1.1B* | *1* | *4* | *2.5* | *~200* | *<20 sec* |
| [2nd LLM] | *1.7B* | *5* | *1* | *3* | *~200* | *~2 min* |
| [3rd LLM] | *1.3B* | *2* | *1* | *1.5* | *~100* | *~1.5 min* |

**Key Observations and Analysis**

**Summarize your key findings comparing the three models**

***Answer*:**

#### **TinyLlama**

* ***Prompt 1****: The model failed to provide a response, repeating the prompt instead.*
* ***Prompt 2****: Gave a structured and creative story. Followed constraints well and explored emotional development and a twist.*
* ***Tokens****: ~180 response tokens*
* ***Time****: <20 sec*

#### **Phi-2**

* ***Prompt 1****: Strong explanation using analogies and prevention techniques. Very clear and accurate.*
* ***Prompt 2****: Did not respond to the story prompt; instead returned unrelated Q&A on robotics.*
* ***Tokens****: ~140 for Prompt 1, ~0 for Prompt 2*
* ***Time****: ~2 minutes*

#### **GPT-Neo-1.3B**

* ***Prompt 1****: Misinterpreted task; generated a technical explanation irrelevant to a high school audience.*
* ***Prompt 2****: Output was fragmented, repetitive, and did not deliver a coherent story.*
* ***Tokens****: ~60–100 total*
* ***Time****: <1 minute*

*Phi-2 demonstrated the strongest factual reasoning, outperforming others on Prompt 1. TinyLlama struggled with technical prompts but generated the best creative response. GPT-Neo showed the weakest overall performance, often misunderstanding or fragmenting the prompt.*

**How would you describe the relationship between Model Size and Performance?**

***Answer*:**

* *Model size correlates with performance — especially in factual and reasoning tasks. However, creativity can sometimes be achieved by smaller models. Phi-2 performed best on factual content, while TinyLlama was surprisingly creative despite its size.*

**What are some potential use cases for different model sizes.**

***Answer*:**

* ***Small Models*** *(TinyLlama, Phi-2): Good for lightweight applications, edge devices, or basic chatbot tasks.*
* ***Medium Models*** *(GPT-Neo): Intended for more versatile uses, but actual performance varies. Best when inference speed matters more than quality.*
* ***Large Models*** *(not tested here): Ideal for technical explanations, deep reasoning, and high-stakes domains.*

**Reflection**

What surprised you most about the differences between the models?

*Answer*:

* *TinyLlama's storytelling ability was impressive for its size. I expected more from GPT-Neo, which struggled significantly despite being larger than TinyLlama.*

What challenges did you encounter when running the models locally?

*Answer*:

* *Running medium models like Phi-2 required optimization techniques like 8-bit quantization. Limited RAM and no GPU made inference slower and restricted me from testing large models like LLaMA-3 70B.*

How might these findings influence your choice of LLM for different applications?

*Answer*:

* *For storytelling or chatbot UX on devices with limited resources →* ***TinyLlama***
* *For factual or educational content →* ***Phi-2***
* *Avoid* ***GPT-Neo*** *unless absolutely necessary, due to reliability concerns.*

**What did you learn from the LLM generated code?**

The code provided to run the LLM from its API was written with the assistance of Claude. When using existing code and/or LLM generated code, take it as a learning opportunity.

Select a line or multiple lines from the provided code such that when you first read it you said “Hmm, not sure what that does” or “I didn’t know you could do that ..” . Perhaps there were no surprises. That’s OK too. Just report “No Insights”

The Code:

*Answer*:

model = AutoModelForCausalLM.from\_pretrained(..., device\_map='auto', load\_in\_8bit=True)

Do a bit of research and describe what you learned.

*Answer*:

* device\_map='auto': *Automatically spreads model layers across devices (e.g., CPU and GPU if available).*
* load\_in\_8bit=True: *Loads models in 8-bit precision, saving memory and making inference feasible on limited hardware.*

1. *Had the same problem with* mistralai *so I end up using* EleutherAI [↑](#footnote-ref-2)