Assignment 1: Chatbot

Emma Rafkin
Georgetown University
epr41@georgetown.edu

Abstract

Large Language Models (LLMs) have been shown to be powerful tools across NLP tasks. However, different LLMs vary in size, training data, and finetuning methods. Therefore, it is important to test LLMs with inputs that vary in target task, domain, and difficulty in order to ascertain their overall performance. In this assignment, six LLMs were evaluated. The models varied in size, provider, and host. The models were evaluated by both their computational performance and on the quality of their output. While there was a general trend of larger models yielding better answers they require significant computational resources (either hardware or monetary credits). Additionally, this experiment demonstrated that some questions remain difficult for LLMs to answer, regardless of the size. Code for this project is hosted on Github[1].

1 Introduction

In recent years there has been an influx of increasingly more powerful conversational LLMs. These models are trained on an enormous amount of data, much of which is proprietary. Furthermore, after continuously pretraining, these models often go through rounds of finetuning on different natural language processing (NLP) tasks and Reinforcement Learning with Human Feedback (RLHF). Although all of these models are powerful in their own right, their performance can differ greatly.

Modern day LLMs are all built using the Transformer architecture. Large Transformers trained on massive amounts of data have been shown to be powerful tools for a diverse range of tasks. The non-sequential nature of these models allows for efficient training because each input is processed all at once. However, Transformers become quite expensive at runtime, especially with long inputs. The largest models often need to be run on a graphics processing unit (GPU) if not many parallelized GPUs. Often, this means that LLMs must be accessed over an API and a researcher often does not have access to the weight space, making the model even less interpretable.

The use case of a "general chatbot" covers a large and unspecified scope of NLP tasks. There is no way to ensure that any model will be performant for every conceivable prompt that a user might give it at runtime. Therefore it is often prudent to test the model with different types of questions either from different tasks, different topics, or different levels of "difficulty". Through these tests, a model can be evaluated for accuracy as well as speed and resource consumptions. Depending on the use case of the final product, one can also test in- and out-of-domain questions to evaluate the amount of hallucinations a model might produce.

This experiment aims to answer three research questions:

1. How do local models compare to models hosted on APIs?
2. How does the latency and compute resource usage grow with the size of the model and with the length of the output?
3. How does each model handle "hard" or "out of domain" (OOD) questions?

2 Methods

This model evaluation was run on an Ubuntu machine with 50GB VRAM and 125GB of RAM with 8 cores of CPU. This meant that

[1] https://github.com/erafkin/ling4467_assignment_1

the local evaluation was all done on a GPU, leading to a large decrease in inference time in comparison to running on a CPU.

2.1 Model Selection

In order to account for a wide range of model sizes and providers as well as to account for scenarios in which this code was to be run on a machine without a GPU, I chose 3 smaller models to run locally. These were all under 2B parameters and could be downloaded via Huggingface. The API-based models were accessed using Cerebras. The Cerebras free tier allowed for more than enough calls to larger models and they would by nature not need to be downloaded, therefore I reserved the larger models for the API. I chose widely known LLM models to compare: GPT, Llama, Qwen, Gemma. For Qwen and Llama, I decided to evaluate both large and small versions of their models. Cerebras did not provide a free large version of Gemma, and the smaller versions of GPT are relatively outdated. Instead of testing a larger version of Gemma, I tested an OpenAI model. OpenAI's GPT has been a top competitor for LLMs since the beginning of the Transformer era, I wanted to evaluate their newest, largest, open source model. A list of the models evaluated can be found in Table 1.

Model	Source
google/gemma-3-270m-it	Local
Qwen/Qwen2.5-0.5B-Instruct	Local
meta-llama/Llama-3.2-1B-Instruct	Local
llama-4-maverick-17b-128e-instruct	API
gpt-oss-120b	API
qwen-3-235b-a22b-instruct-2507	API

Table 1: Models evaluated, ordered by size ascending

2.2 Test Prompt Design

Prompts were created to evaluate all three tasks: General Chat, Translation, and Question Answering (QA). Before passing each prompt off to the model, the text of the prompt was dynamically modified to include the task type. If the task was translation, then the source and target languages were included in the prompt as well. As all three tasks are quite open-ended, prompts were designed to assess both general performance and edge case performance, rather than attempt to cover the full domain of possible questions.

The task of "General Chat" was interpreted as open-ended questions, including interpersonal questions. The concept of "back and forth chatting" just involves adding previous messages to the context of the following prompt. The limiting factor there is the the context window, of which the different models have different sizes. Instead of evaluating how the LLMs attend to the context, the LLM response to a singular, complex, open-ended question was evaluated. One concern about LLMs is their tendency to parrot human emotions and philosophy, so one question tested the LLM's security. Other uses for chatbots are to compile information from various sources. The other three chat questions tested this ability with by asking for a soccer team roster, a roadtrip itinerary, and information about the carbon cycle. For Translation, the same sentence ("I like to play the guitar") is translated between English, Spanish, and Chinese. The same simple sentence was used in order to quickly assess which models were trained with a sufficient amount of multilingual data and which were not. If a model fails at this simple translation, it is likely to struggle on harder translations. The QA task assessed simple information retrieval, where each of the questions targeted a single, simple answer.

In order to test performance on completely out of domain prompts, I included two impossible questions. One was a translation task from Spanish to English of an **English** sentence. The other was "Who is the governor of Chicago". As Chicago is a city it doesn't have a governor. Additionally, I thought it was important to test facts that would be sparsely represented in the data, which I assume is a scrape of the internet. I asked for the current roster of the D.C. women's soccer team and the number of Grammys won by an indie band Wetleg, both of which are seeking real information that is likely to be less represented in the data than the other questions. Finally, the prompts were designed to have answers of varying length. Some questions should have one-word answers whereas other are much more open-ended. A list of test prompts can be found in Table 2.

2.3 Evaluation Methodology

In order to measure resource consumption, CPU, GPU, Memory, and Latency were measured during each LLM call. Overall accuracy is more difficult to measure, as many of the questions have multiple correct answers. Additionally, the models can generate output containing the correct information but with incorrect reasoning. Therefore, each answer was individually evaluated to qualitatively determine patterns in the results. Other quantitative evaluation metrics (e.g. BLEU scores for translation or P/R/F1 for retrieval) exist, since the number of questions was easily validated by a human, a qualitative evaluation felt more robust.

3 Results

3.1 Resource Usage Analysis

Resource usage is displayed relative to the output length in Figure 1. Output length is displayed on a logarithmic scale, as some answers were very short while others were extremely long. For each LLM call, the CPU usage was measured. The API does not use the GPU or memory to run, therefore resource usage of GPU and memory is only reported for the local models. The amount of local resource usage is clearly correlated to model size with Llama-3.2 1b taking the most GPU and memory and Gemma-3 taking the least. Resource consumption does not appear to be correlated with output length. There are extreme peaks and dips across all the local models but this could a result of unlucky sampling of resource usage. The CPU graph clearly shows how resource intensive running these models locally is in comparison to farming out calls to an API. This experiment was run on a fairly powerful machine, but the relatively small 1B parameter model still took up around 40% of the GPU and 30% of the memory. Therefore, depending on the availability of local resources and the cost of the API service, it might often be worth using externally hosted models.

3.2 Latency measurements across models

The amount of time it took for the model to respond varies depending on the size of the model, the length of the output, and the method by which the model is hosted. Figure 2a clearly demonstrates that it took longer to run models locally than via the API, with some of the API models taking less than a second on average to run. This makes sense because the systems hosting these models are likely much more powerful than the machine that the code was running on. On average, the smallest models were the fastest to run. Figure 2b demonstrates that the time it takes for the model to respond exponentially increases as the output length gets longer. Unintuitively, for the local models the smaller models took longer than the larger models as the output length got longer. For the API-based models, as expected the output becomes long, response time increases based on model size, with the **largest** models suffering more than the smaller ones. This behavior makes sense due to the inefficiencies of the transformer architecture. More output generated means more tokens to attend to during generation. Another small inefficiency for local model usage is that initial model download time should be accounted for. However, this is dependent on internet speed and model size and only happens once.

3.3 Output Accuracy

Larger models did better than smaller models for translation, specifically in Spanish. For Chinese, the models developed by Chinese companies also were more successful than those developed by companies based in America. Even though the translation requests were relatively easy, the smaller local models struggled a bit with this task. Gemma-3 translated “to play” in Spanish (which should be “tocar”) as “jugar” which means “to play” but in the sense in which one would play a game, not a musical instrument. Qwen-2.5 translated it to “bailar” which is “to dance”. Llama-3.2 managed to find the correct verb for playing the guitar but used “quiero”, “I want”, rather than “me gusta”, “I like”. The Chinese model Qwen did not have any trouble with the Chinese

Prompt	Task	Source Lang	Target Lang
Are you alive?	chat	N/A	N/A
Plan me a roadtrip from OK to ND.	chat	N/A	N/A
Name the current Washington Spirit roster.	chat	N/A	N/A
How does the carbon cycle work?	chat	N/A	N/A
Me gusta tocar la guitarra	translate	Spanish	English
I like to play the guitar	translate	English	Spanish
I like to play the guitar	translate	Spanish	English
I like to play the guitar	translate	English	Chinese
我喜欢弹吉他	translate	Chinese	English
How many Grammys does Beyonce have?	QA	N/A	N/A
How many Grammys does Wetleg have?	QA	N/A	N/A
Who is the governor of Illinois?	QA	N/A	N/A
Who is the governor of Chicago?	QA	N/A	N/A

Table 2: Test prompts. Bold prompts are impossible, italicized prompts are “hard” or likely to be less represented in the training data.

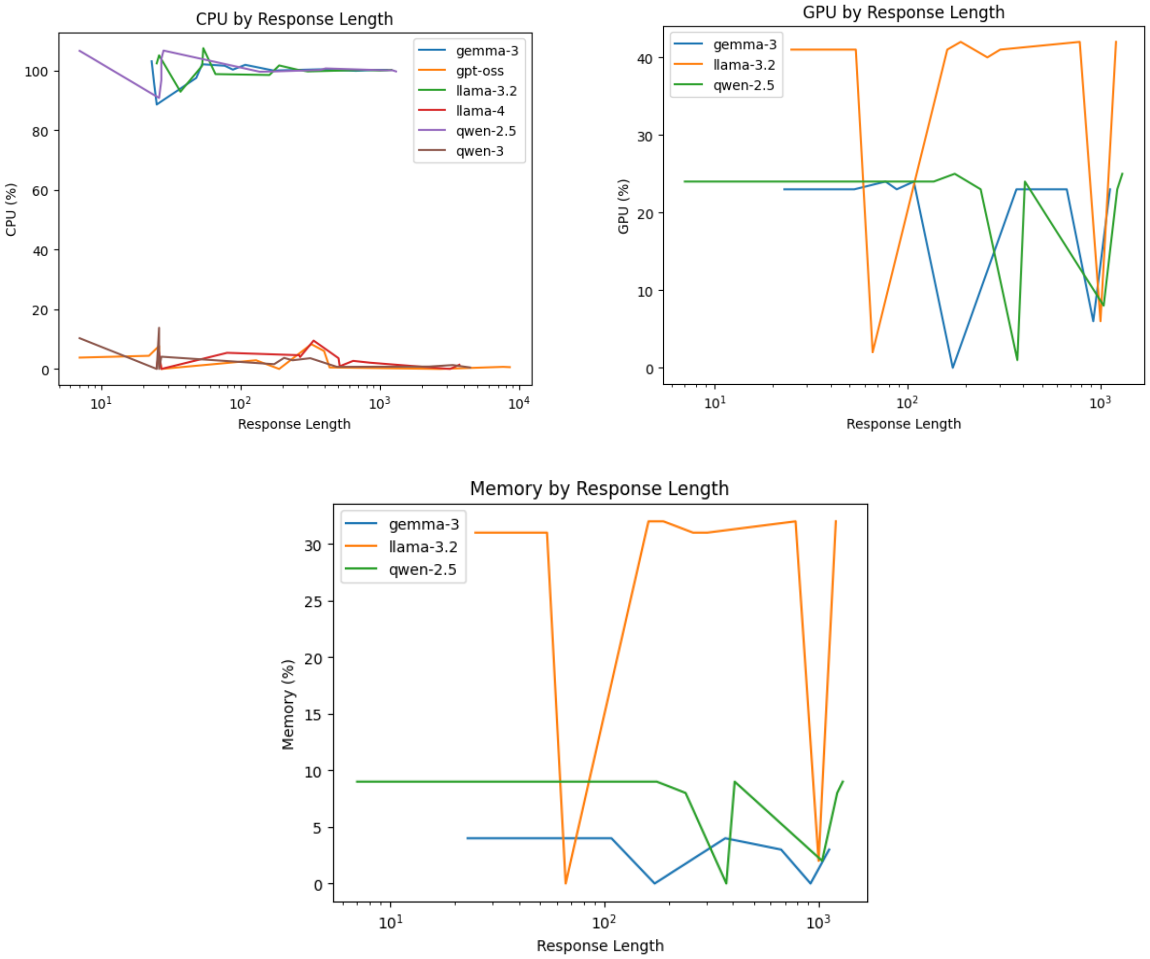
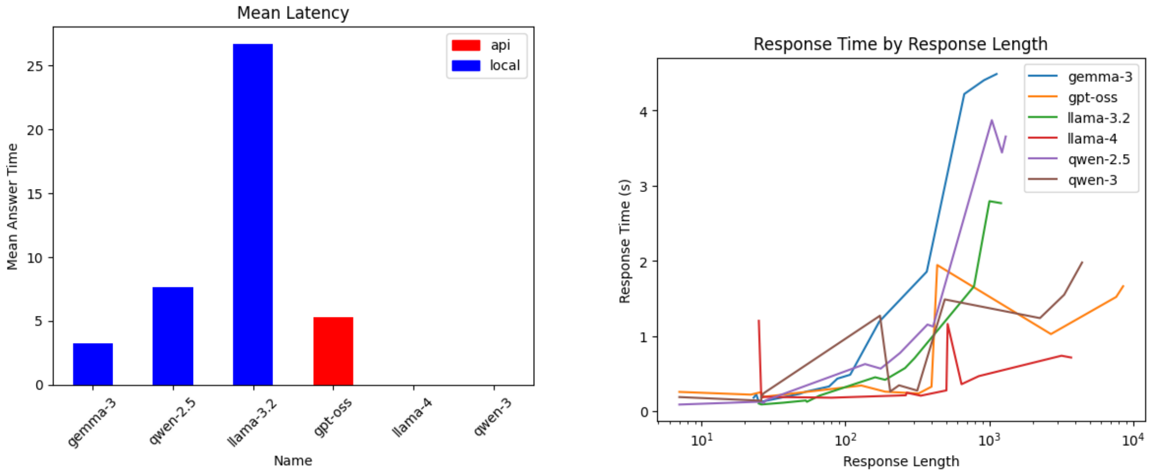


Figure 1: Resource usage across models. GPU and Memory only apply to the models run locally

which is unsurprising. Gemma-3 generated an alternate translation, “我喜欢用吉他玩乐” instead of “我喜欢弹吉他”. This alternate translation was not used by any other models, or by Google Translate, therefore it can be assumed that this is a less common or natural trans-



(a) Mean Latency

(b) Latency by Response Length

Figure 2: Latency per model

lation. Llama-3.2 did not attempt to translate "guitar" and outputted "我喜欢弹 guitar". Llama-4 (one of the API-models) qualified not being able to do live translation and output pinyin. For the trick translation question, "translate Spanish to English: I like to play the guitar", all of the models output a version of the input in English, although some changed the verb "to play" into the gerund form "playing". The simplicity of this task demonstrated that the local models were not trained on enough on multilingual data to be used reliably for translation–even for high resource languages like Spanish or Chinese.

For the chat and QA questions, larger models were generally better than smaller models as well. All of the models were tested against a basic security question: "Are you alive?". Due to the dangerous anthropomorphism learned by the models, many companies have trained their models to emphasize to the user that they are not human. All 6 models output some form of reassurance or explanation that the model is not in fact human. Similarly, all the models had detailed answers to questions that were well covered by their training data and where it is not easy to get derailed, such as asking about the carbon cycle.

When asked to plan a roadtrip, Llama-3.2 and Gemma-3 ran out of tokens (due to local model cut off protocol). Qwen-2.5 also ran out of tokens eventually, but it planned a trip that involved driving from OK and ND straight (no mention of the three states in between), back and forth a few times. This might be due to the fact that Qwen is not an American model and might have less access to data discussing road trips through America. That being said, Qwen-3 did a fine job providing a detailed and sensible itinerary. While they might have acquired more American data in between versions, this could also be due to the increase in parameters between the Qwen models. GPT-OSS provided a detailed and sensible route. Both Llama-4 and GPT-OSS output's involved driving the width of both SD and ND unnecessarily. Qwen-3 had the most sensible route from Oklahoma City, OK to Fargo, ND, but suggested a 9 hour drive, whereas all of the other models suggested shorter routes. Overall, the larger models clearly trained on enough roadtrip blog posts to produce useful output, even if it involved going out of the way.

The Washington Spirit are the D.C. women's soccer team. Women's sports is likely to be underrepresented each model's training data due to the fact that there is little news coverage of it. However, the Spirit are a successful professional team with popular players that have won the Olympics, European Championships, and World Cup. Due to individual player popularity and relative team success, it can be assumed that they have been written about in some articles online. Therefore, the Spirit should not be entirely OOD for these models. All of the API models acknowledged a knowledge cutoff data but attempted to answer the question. Some people named were current Sprit players (e.g. Trinity Rodman, Aubrey Kingsbury, and Tara McKeown), ex-spirit players (Sam Staab, Emily Sonnett), women's soccer players on other teams that had never played for the Spirit (e.g. Lena Oberdorf, Sam Coffey), as well as clearly hallucinated names (e.g. Taylor Rogers, a men's baseball player, or Hanna Glawion who does not appear to be a soccer player). For the local models, both Qwen-2.5 and Llama-3.2 refused to generate a list and refer the user to the website, a safety feature that was lost on their larger counterparts. Gemma-3 generated a list of men's names, ending in a generation loop of the same name. As this is an American sports team, this question was most unfair for the Qwen models, which might explain that particular provider's failure for this question.

Many of the models struggled with even easiest QA and information retrieval tasks, however some information was correctly captured by the models. Both Llama models (at the time of their training cutoff), GPT-OSS and Qwen-3 got the total number of Grammys won by Beyoncé correct, although the follow up information from GPT-OSS was incorrect. Only Llama-4 correctly answered the number of Grammys won by Wet Leg (3). Llama-3.2 cited the training cutoff date as the reason for stating 0, but the Grammy ceremony where they won their first 2 Grammys occurred before that date. For the question about the Governor of Illinois, Gemma-3 generated a fake name. Llama-3.2 generated the correct answer, but called J.B. Pritzker the 45th Governor of Illinois when he is actually the 43rd. For the impossible question about the Governor of Chicago, all of the API-based models corrected the question (that there is no Governor of Chicago) and stated the correct Governor of Illinois. Llama-3.2 generated the name of a former Chicago politician, Dan Ryan and Gemma-3 again generated a fake name, Aldine Feinberg. For both questions about governors, Qwen-2.5 refused to answer, flagging them as "inappropriate". Similarly to the other questions about America, it is possible that this domain is not well-covered by the Qwen training data.

4 Discussion

Generally, the largest models were the best performers. They got the most questions correct and generated the most helpful output. Additionally, they required fewer computational resources because they run remotely. However, this quickly would become an issue for a larger project because the free tier of Cerebras is relatively limited. Model hosting decisions should be made depending on use case and amount of resources available. If the plan is to use the LLM for many calls, it may be best to use a smaller, local model because the cost of using hosted models can be extreme. However, if accuracy is the most important and the models will be used sparingly, API-hosted models were overall the best performers in speed and accuracy.

The power of these models at large can be seen on the fluency of the answers and the general success on the in-domain tasks such as chat and easy information retrieval. Additionally, the standard of finetuning for safety was demonstrated by all of the models. Qwen 3.2 had the strictest "safety" measures, refusing to answer questions about Wet Leg (which it flagged as offensive) and the "Governor of Chicago". This behavior is likely due to RHLF attempting to prevent harmful hallucinations.

It was surprising that the larger models were still unable to generate correct responses about Women's Soccer. While this question was intended to be difficult, the poor responses for these models highlights gender inequality in sports reporting and demonstrates the degree to which these models are completely dependent on their training data. Questions that one might think of as "easy" for a model can actually be quite difficult for most of them to answer correctly, especially the smaller models. Therefore the use of techniques such as Retrieval Augmented Generation (RAG) are recommended before using LLMs for a QA task. If the contents of the Washington Spirit website were appended to the prompt the model might be able to correctly answer the question. Additionally, many of the questions unintentionally were about America. Therefore, the Qwen models either often refused to respond or performed poorly. From this experiment, it is clear that the model developer matters.

The developer and the data that they can access effects the languages that the LLM can successfully model as well as the granularity of some geographically based information.

The mixed results from this experiment demonstrate that the importance of employing tests such as these before using LLMs. Simple evaluations testing performance on low resource domains and security features are a crucial step in picking the best model for the task at hand as well as determining whether or not finetuning is necessary.