A Cost Efficient, LLM Routing System

Rayan Ghabashi, Abdulqader Bariah, Iyas Baroud, Hamza Alkaf, Mohammed Mukhtar, Abdulrahman Saad

ABSTRACT

In this project, we offer a technique for querying LLMs while maximizing quality, speed, while maintaining minimum cost. First, a semantic cache is established to handle frequent prompts, saving time and money. Then, a router directs simple prompts to low-cost local models like llama, further maintaining the cost. Lastly, if the prompt is set to be urgent by the user, it is passed to a powerful, fast, but costly model such as GPT4o. If the request is not urgent, it is routed to a powerful free model, such as Mixture of Agents, which produces high-quality but slower results. In this study, we discuss our system's architecture as well as the results, which demonstrate the tremendous benefit of combining such basic components. Finally, the paper concludes with citations to additional resources.

INTRODUCTION

Large Language Models (LLMs) are advanced artificial intelligence systems designed to understand and generate human-like text based on vast amounts of data. They are trained on varied datasets, which frequently include billions of words from books, papers, websites, and other sources, allowing them to perform a wide range of language-related tasks. These responsibilities include answering inquiries, writing essays, translating languages, summarizing material, and even producing creative literature such as poetry and fiction.

GPT-4o, also known as GPT-4 Open, is an improved iteration of OpenAI's Generative Pre-trained Transformer (GPT) series and serves as an example of an LLM. GPT-4, which builds on earlier versions' accomplishments and lessons learned, is intended to improve language models' capacity to read and generate human-like writing. In addition, GPT-4o has enhanced abilities in specialized tasks like translation, summarization, and even coding. Its architecture enables it to interpret and write text in different languages, making it an important tool for global communication and cross-cultural interchange.

However, when attempting to integrate AI in a system, developers resort to GPT4o as their only tool which is inefficient in terms of cost and speed. Even though many prompts might aim for the same response, each query generates a distinct response. Furthermore, in many circumstances, most queries are basic and do not require a powerful model like GPT4o to handle them, keeping in mind that GPT4o charges $5 per 1 million tokens.

PROPOSED IDEA

The problem is heavy reliance on GPT4o for everything. We attempt to keep costs down while maintaining quality and quickness. Technically, there are three aspects to improve in this problem:

The first is repeated queries. Those may be handled with a semantic cache, which compares query similarities and pulls them from the cache if they have already been answered, so they do not have to be regenerated each time a query is received. The second aspect is simple queries that do not need GPT4o. A decent option would be to route those to a weak model, saving the time and money of sending them to GPT4o. Lastly, we need a free alternative to GPT4o that may maintain acceptable quality, even if it means compromising speed since a corporation might be out of budget and cannot afford to bring the entire system down due to GPT4o.

SEMANTIC CACHE

a sophisticated caching method that caches previously created responses to queries based on their semantic content rather than the specific syntax of the query. When an unfamiliar query is sent to the LLM, the system first evaluates its semantic meaning, which includes understanding the content rather than just the exact phrase. If a semantically comparable response is located in the cache, the system fetches the previously stored response.

LLM ROUTER

An LLM router is a simple classification model that rates the complexity of the query on a scale of 1 to 5. If the prompt is rated above 3, it is classified as complex. This router has been pretrained on prompts and their corresponding responses from both a strong and a weak model. The strong model was then asked to rate the responses according to their sophistication, creating a dataset that the classification can be trained on.

MIXTURE OF AGENTS

Mixture of Agents or MoA is an approach that leverages the diversity of multiple LLMs or specialized components within an LLM ecosystem, allowing for more tailored and effective solutions to complex language tasks. By combining the strengths of several agents, this strategy can overcome the

constraints of single-model systems and deliver a more powerful and adaptable AI-driven tool. These agents are arranged in layers; each layer contains multiple agents that receive the same prompt, and their results are combined and passed to the next layer. At the end of this architecture resides an aggregator LLM whose job is to combine all responses into a final coherent one. This architecture however is relatively slow to respond.

SYSTEM ARCHITECTURE

Combining all the aforementioned, we get the following:

Our system initially checks to see if the response is cached; if it is, it is fetched and returned. The complexity check comes next,, where simple prompts go to a local weak model such as llama3. Finally, the urgency condition is set by the user; if the prompt is not urgent, it goes to MoA, where it takes its time; if urgent, it is forwarded to GPT4o as a last resort.

CONCLUSION

In conclusion, our study presents a novel and effective approach to optimizing the use of large language models (LLMs) by strategically combining various techniques to balance quality, speed, and cost. The use of a semantic cache to handle frequent prompts saves substantial time and money, while the use of a routing mechanism guarantees that resources are allocated efficiently based on the complexity and urgency of the query. By directing simple prompts to low-cost local models like Llama, urgent prompts to powerful but costly models like GPT-4o, and non-urgent complex prompts to free but slower models such as Mixture of Agents, we have shown that it is possible to maximize the efficiency of LLM usage without compromising on output quality. This research underscores the importance of thoughtful design in AI system architectures and opens avenues for future exploration in the field of cost-effective AI deployment.

REFERENCES

https://arxiv.org/abs/2406.04692 https://arxiv.org/abs/2406.18665