Optimizing Tool-Based Query Solving Using Fine-Tuned Language Models: A Comparative Study of LLaMA, GPT, and Groq Models

Om Shende, Rahul Vimalkanth, Sachin

Abstract

This paper presents a comparative study of tool-based query solving using fine-tuned Large Language Models (LLMs), focusing on LLaMA, GPT, and Groq models. The work is part of the InterIIT Hackathon 2024, where the objective was to leverage LLMs to automate complex query solving through tool calls. By employing a two-stage LLM architecture, this study demonstrates how LLaMA and Groq's LLaMA3 models, and GPT-based Few-shot Chain of Thought (CoT) prompting, integrate with predefined tools to execute tasks like database querying, summarization, and problem-solving. The evaluation shows high accuracy, low latency, and the effectiveness of each model's prompting techniques. Challenges such as hallucination and tool argument handling are discussed, along with the potential improvements in future iterations.

1. Introduction

Large Language Models (LLMs) have shown significant advancements in handling tasks across multiple domains, particularly in natural language processing and tool-based query solving. Recent developments, including Groq's fine-tuned *LLaMA3* model and GPT's Few-shot Chain of Thought prompting, offer powerful solutions for structured tool-based environments.

In this work, we aim to explore and compare these different LLMs for solving predefined queries using external tools. As part of the InterIIT Hackathon 2024, the goal was to implement models that could select and call tools effectively based on the user query while adhering to a specific JSON schema. The models were tasked with generating accurate tool calls, optimizing latency, and handling complex real-world tasks. This paper discusses the architecture, methodology, and performance of Groq's LLaMA3, LLaMA 3.18b, and GPT-4 models in tool-based query solving.

2. Related Work

Recent studies have demonstrated the use of LLMs for function calling and task automation. Models like LLaMA and GPT-4 have been shown to improve the efficiency of task execution by integrating external tools. While Retrieval-Augmented Generation (RAG) and structured prompting approaches have emerged as key techniques, challenges persist in handling multi-step queries and tool argument dependencies.

Our work builds on this foundation by focusing on three specific models: Groq's fine-tuned LLaMA3, GPT-4o, and the LLaMA 3.18b models. We explore how these models perform in constrained environments, such as the InterIIT Hackathon, where tool use is central to solving predefined tasks.

3. System Architecture

3.1. Groq's LLaMA3 Model

Groq's fine-tuned *LLaMA3-Groq-70B-8192-tool-use-preview* model is specifically designed for tool calling. It operates using a two-model architecture:

3.2. LLaMA 3.18b with Few-shot Chain of Thought Prompting

The LLaMA 3.18b model was employed using Few-shot Chain of Thought (CoT) prompting, which introduces intermediate reasoning steps. The model's prompt includes comprehensive tool descriptions and examples, guiding the model to use tools efficiently and correctly pass outputs between tool invocations.

Key elements of the prompt:

- Tool Descriptions and Arguments: Detailed instructions help the model understand the required inputs and outputs of each tool.
- Sequential Tool Invocation: The model is prompted to compare tool descriptions and correctly chain tools together.
- Authentication Check: Ensures that necessary tools like `who_am_i` and `team_id` are used for proper authentication.
- Fail-safe Mechanism: If no available tools can solve the query, the model returns an empty list.

3.3. gpt-4o-2024-08-06: Few-shot Chain of Thought Prompting

GPT-4o's CoT prompting uses a similar strategy, where structured intermediate steps break down complex queries into manageable tasks. In this approach, the model is trained to recognize when tools are needed and how to invoke them in the correct order. The Few-shot CoT technique provides the model with examples to improve tool chaining, reducing the risk of hallucination and improving accuracy.

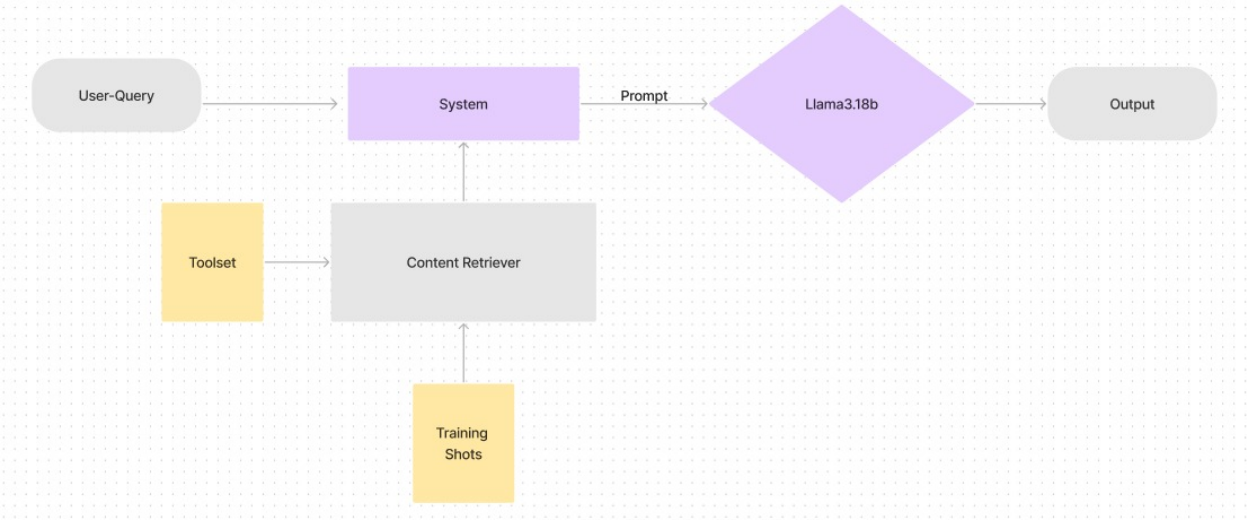

Figure 1: Flowchart of the gpt4-o and llama Model Architecture

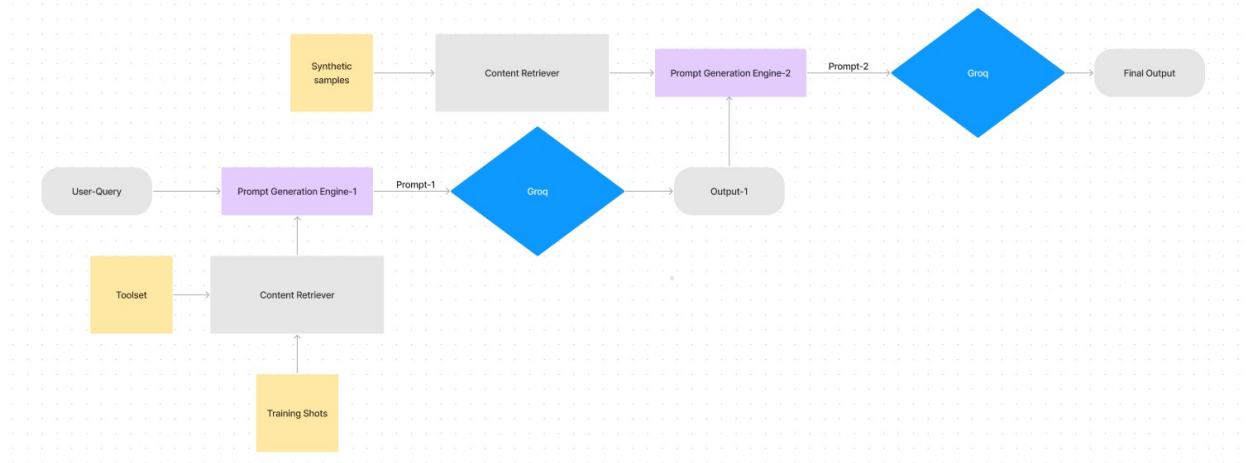

Figure 2: Flowchart of the Groq Model Architecture

4. Methodology

4.1. Prompting Strategy

All models rely on a robust prompting strategy to handle tool-based queries. For Groq's LLaMA3, MODEL1 generates tool calls based on basic queries, while MODEL2 refines these calls using synthetic data. In LLaMA 3.18b and GPT-4o, Few-shot CoT prompting is employed, which guides the models step-by-step through tool selection and chaining.

For complex tasks involving multiple tools, we generated synthetic datasets requiring sequential tool usage. In these cases, each query referenced previous tools' outputs, allowing the models to handle complex scenarios more effectively.

4.2. Evaluation Metrics

Performance metrics for all models were measured using ROUGE and BLEU scores. The ROUGE-L F1 score, which measures the longest matching subsequence between the generated and expected results, was particularly useful in evaluating the models' accuracy in tool chaining. The test dataset for carefully crafted to include single tool, multi-tool and no-tool queries for rigorous testing of the models.

5. Results and Discussion

5.1. Groq's LLaMA3 Performance

Groq's LLaMA3 model demonstrated strong performance, achieving ROUGE scores between 60 and 80 across multiple test cases. Latency was low (approximately 6 seconds per query), although it increased for more complex queries. The model's ability to generalize from few examples allowed for accurate tool selection, although hallucinations occurred when dealing with overly complex tool sequences.

5.2. LLaMA 3.18b with CoT

LLaMA 3.18b, using Few-shot Chain of Thought prompting, outperformed the Groq model in handling complex multi-tool queries. The intermediate reasoning steps introduced by CoT improved accuracy and reduced hallucination, though latency was slightly higher due to the increased reasoning complexity.

5.3. GPT-4 Few-shot CoT Performance

The model was given a prompt that spoon-fed the expected thought-process of the model. This technique was found to give a more relevant answers than zero-shot prompting . The prompt was optimised to produce the best results at a low cost. The tools were provided in JSON format, with careful emphasis on certain arguments. The output mode was specified to be a JSON object. The model was made to generate synthetic data using a customized prompt for use during evaluation. The latency is around 2-3.5 seconds depending on internet speed.

The BLEU and ROUGE-L scores of GPT-4o were higher than the other models. Very little to no hallucination was observed.

BLEU score: 0.75 ROUGE-L F1: 0.94

The model aced single tool, multi-tool and no-tool prompts. It was observed that it made errors only when the language of the query was ambiguous enough to cause confusion between arguments that are semantically very similar. There was sometimes a mismatch between the order of arguments in the truth value and the model's response which lowered BLEU and ROUGE-L scores a little bit, but has no real significance as a "mistake". The performance/cost ratio of GPT-4o was found to be higher than other popular paid models.

6. Challenges and Limitations

Despite their strong performance, the Groq and Llama models faced challenges with hallucination, where they returned excessive tool calls or incorrect arguments. While the use of two chained models in Groq's

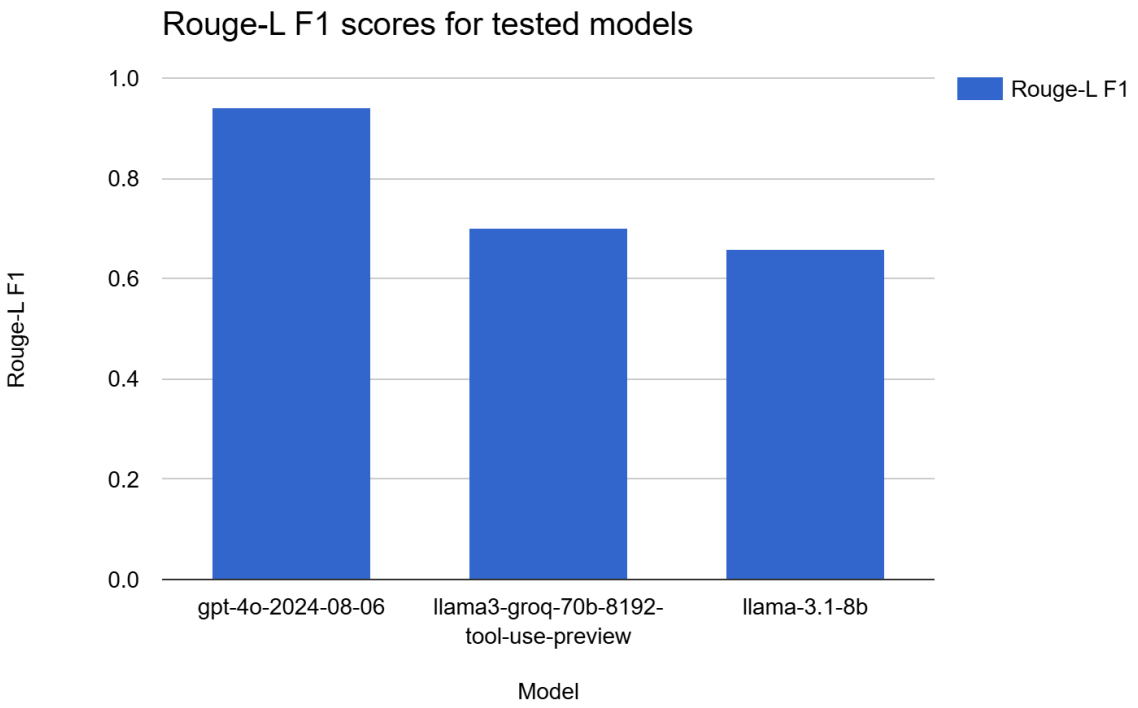

Figure 3: Evaluation comparison

architecture helped reduce errors, further refinement is needed to handle complex queries without hallucination.

Another limitation was the difficulty in handling real-time tool calling. The latency for complex queries could be reduced by optimizing the model's understanding of tool dependencies. Also, GPT-4o is a paid model with a price of 2.5USD/1M tokens.

7. Conclusion

This study demonstrates the potential of fine-tuned LLMs, including Groq's LLaMA3, LLaMA 3.18b, and GPT-4o, in solving tool-based queries. The two-model architecture of Groq's system, coupled with few-shot Chain of Thought prompting in LLaMA and GPT-4o, enables efficient and accurate query resolution in structured environments. However, challenges such as hallucination and latency in some models highlight areas for future work. Overall, GPT-4o outperformed the other models in terms of BLEU score, ROUGE-L score and latency.

8. Future Work

Future efforts will focus on:

- Implementing a full Retrieval-Augmented Generation (RAG) system to improve example retrieval during tool selection.
- Reducing latency for complex queries by optimizing the models' understanding of tool arguments and dependencies.
- Addressing hallucination through more refined prompting strategies and task-specific fine-tuning.