UNIVERSITY OF MICHIGAN

Automated Task Scheduler And Calendar Assistant

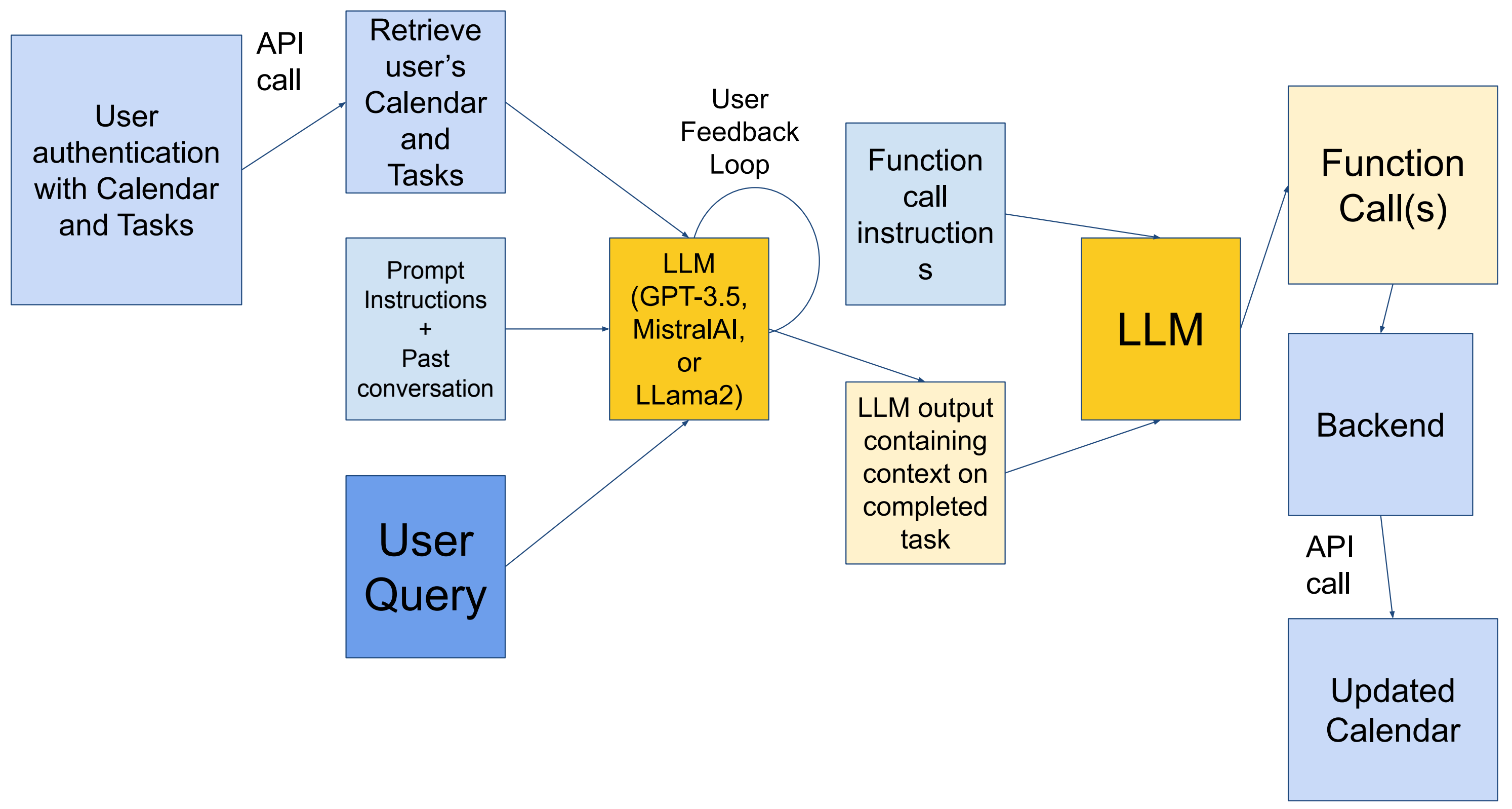
Shriya Ejanthker, Marissa Bhavsar, Xinhe Wu, Ojaswi Bhimineni

Motivation

Modern workplaces demand efficient time management and scheduling solutions. Virtual assistants, powered by large language models (LLMs) like GPT-4, Mistral, and Llama2, offer promising avenues to automate these tasks. However, ensuring these assistants operate with high accuracy, relevance, and user satisfaction is crucial. This problem has led us to develop an Automated Task Scheduler and Calendar Assistant (ATSCA) by leveraging LLMs for natural language processing and interfacing with the Google Calendar and Google Tasks API to automate scheduling. This project investigates the performance of different LLMs in managing calendar tasks, identifying strengths and weaknesses, and iteratively improving their responses. Our motivation stems from the potential of LLMs to interpret user queries and facilitate human-like interactions, making them promising candidates for automating scheduling tasks.

Methodology

Working Principle


User authentication with Calendar and Tasks

API call

Retrieve user's Calendar and Tasks

Prompt Instructions + Past conversation

User Query

User Feedback Loop

LLM (GPT-3.5, MistralAI, or LLama2)

Function call instructions

LLM output containing context on completed task

LLM

Function Call(s)

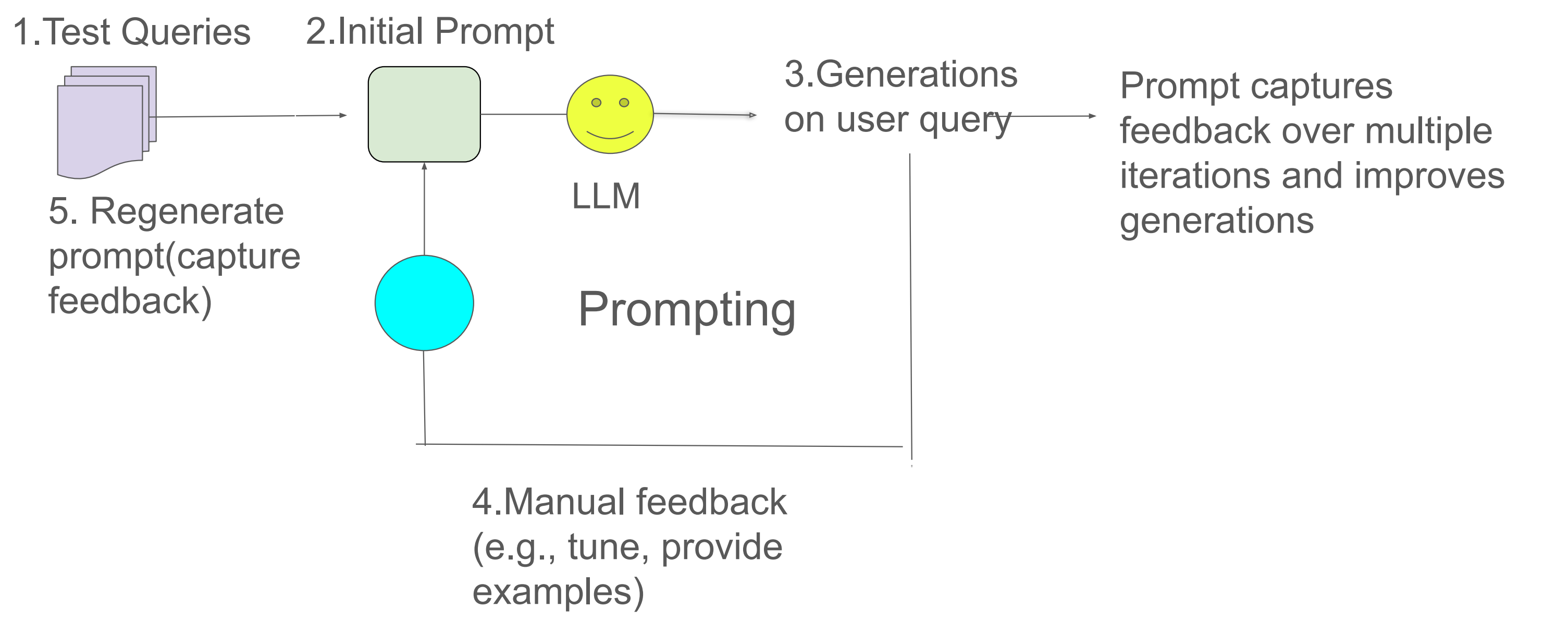
Backend

API call

Updated Calendar

- Model Integration: Incorporated GPT-3.5, Mistral, and Llama2 into a virtual calendar management system, interfacing with Google Calendar API.
- Response Generation: Configured systems to handle scheduling and task management queries, generating responses based on user inputs.
- Iterative user feedback loop and continuous learning: Established a user feedback loops that enables the LLM to ask for clarifications and change responses based on feedback. Also learns from previous conversations with user.
- Function calls: Configured system to call functions that interact with Google Calendar and Tasks API and make necessary changes.

Iterative Prompt Engineering


1.Test Queries

2.Initial Prompt

LLM

3.Generations on user query

Prompt captures feedback over multiple iterations and improves generations

5. Regenerate prompt(capture feedback)

Prompting

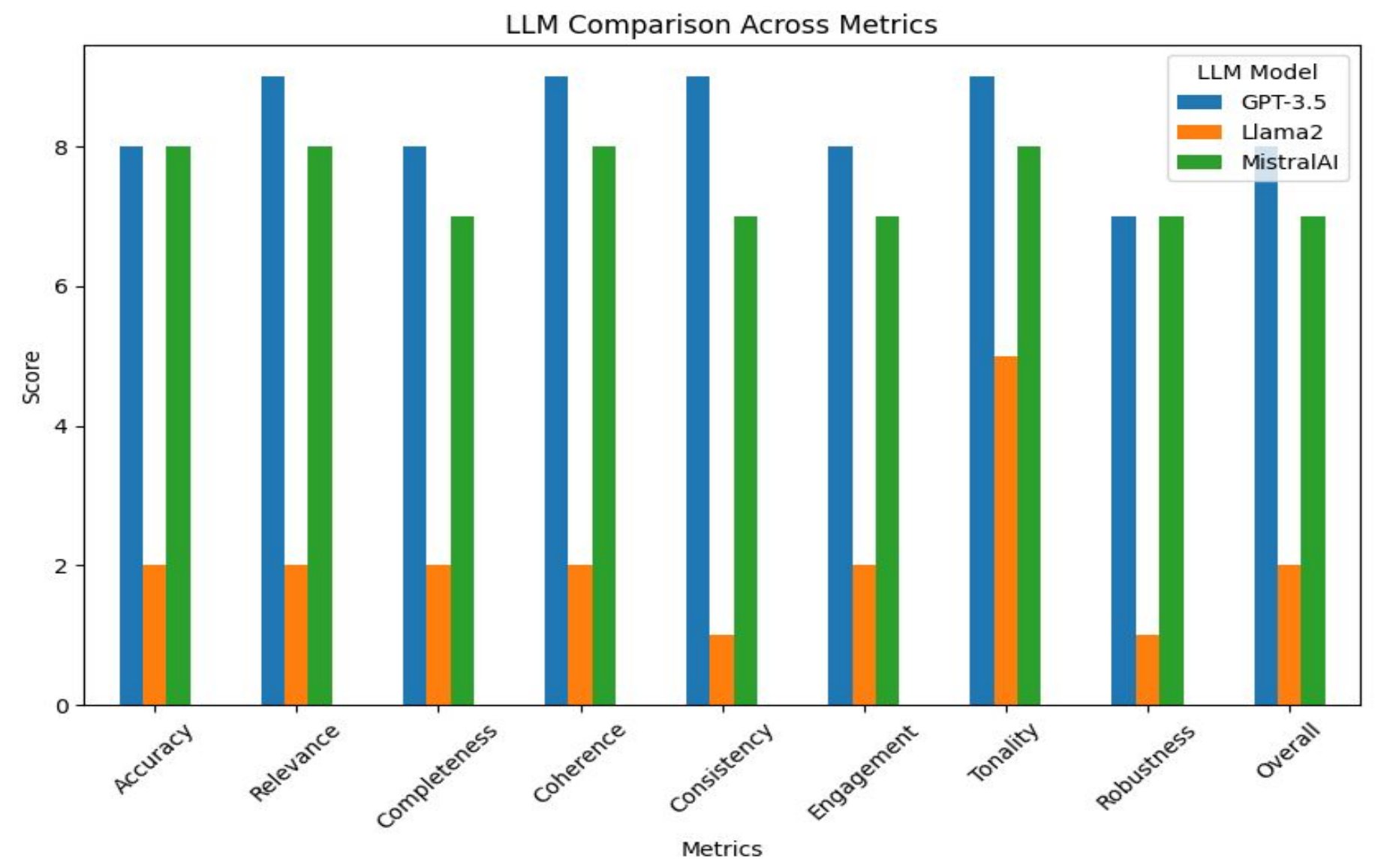
4.Manual feedback (e.g., tune, provide examples)

- Feedback Loop: Implement a feedback loop where responses from the LLM are evaluated based on user satisfaction and relevance. This feedback is used to fine-tune the prompts and response strategies, enhancing the LLM's performance over time.
- Prompt Optimization: Optimize prompt based on feedback to maximize clarity and accuracy of responses.

Evaluation

Table 1: Quantitative Evaluation

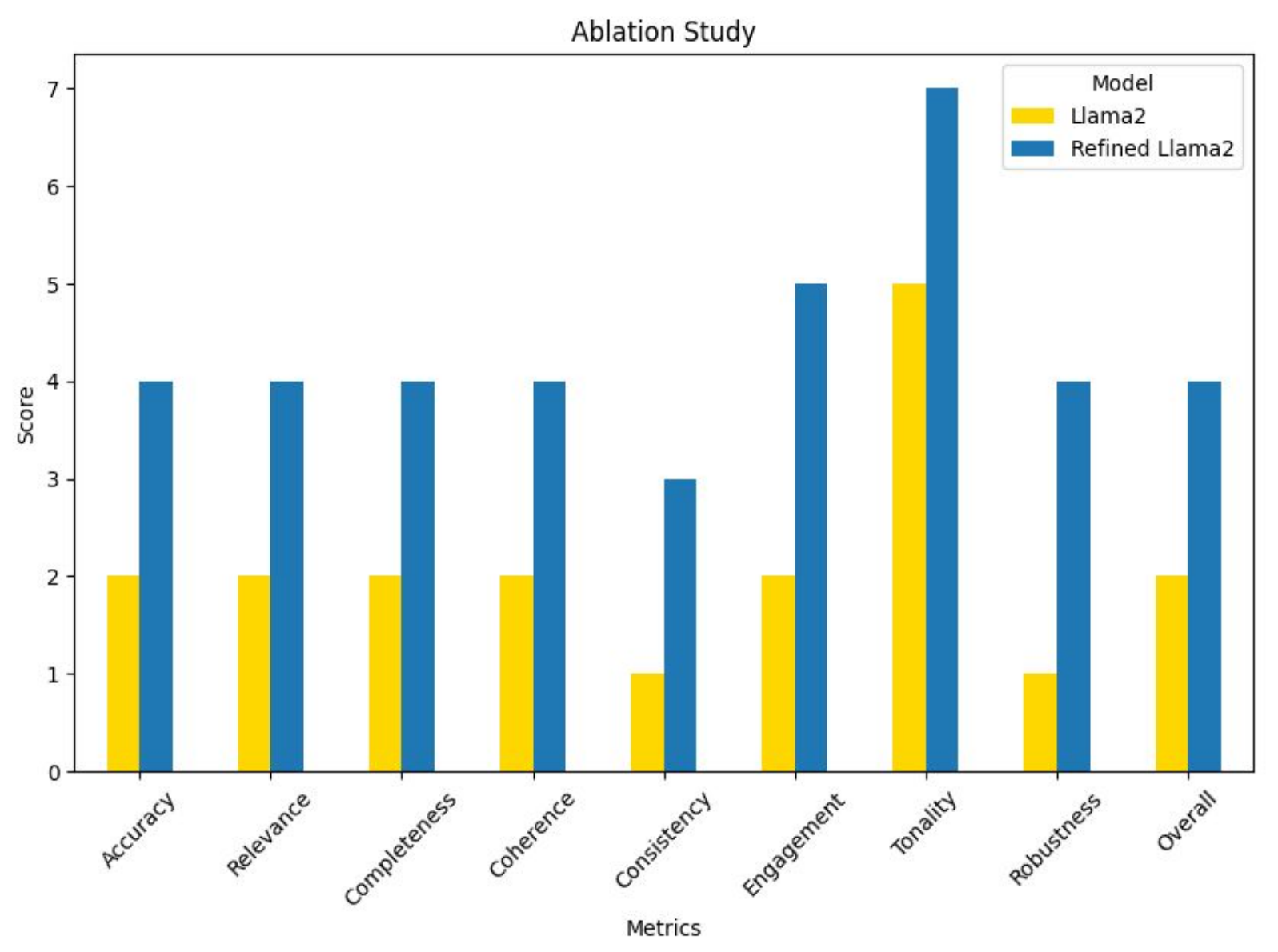
	GPT-3.5	llama2	MistralAI
F1 score	0.844	0.502	0.534
Meteor Score	0.598	0.421	0.455
Valid Response Score	1.000	1.000	0.563



Functional correctness: We conducted a detailed analysis focusing on the accuracy of function calls generated by each model. Using metrics such as METEOR score, F1 score, and valid response score, we quantitatively assessed the correctness of the API function calls initiated.

Intrinsic Evaluation: We also used GPT-4 to evaluate the model's conversation with the user on more user-centric metrics revealed GPT-4's superiority in accuracy, relevance, coherence, engagement, and robustness. In contrast, Llama2 and Mistral showed significant shortcomings in understanding and interacting with users, struggling with completeness and consistent quality.

Ablation Tests



The ablation tests yielded significant enhancements in Llama2's responses. By refining input contexts and narrowing the scope of tasks given, we observed marked improvements in almost every metric for LLama2.

Conclusion

- Initial findings show that while all models handle basic tasks well, iterative prompt engineering greatly improves their performance, especially in robustness and relevance.
- These results support using advanced LLMs like GPT-4 for complex user interactions and tasks.
- The effectiveness of ablation testing underscores the significant benefits of targeted prompt engineering in enhancing the practical utility of language models.

Future Work

- Advanced prompt engineering: Investigate deeper linguistic structures to further refine prompt, experiment with context-sensitive prompts
- Explore hybrid models that combine two or more LLMs and perform a comparative study.