

Introduction

Modern navigation apps are extremely effective for cars, but are missing the adaptability necessary for bicycling. In this work, I have attempted to address these shortcomings by incorporating a *Large Language Model* (LLM) into the standard navigation app structure through the use of Few-Shot Prompting.

Objectives

- Develop a solution for more effective bicycle navigation.
- Learn about LLM prompting techniques and system design strategies.
- Compare effectiveness of open-source LLMs vs. commercial models (ChatGPT), as well as single models vs. ensemble.

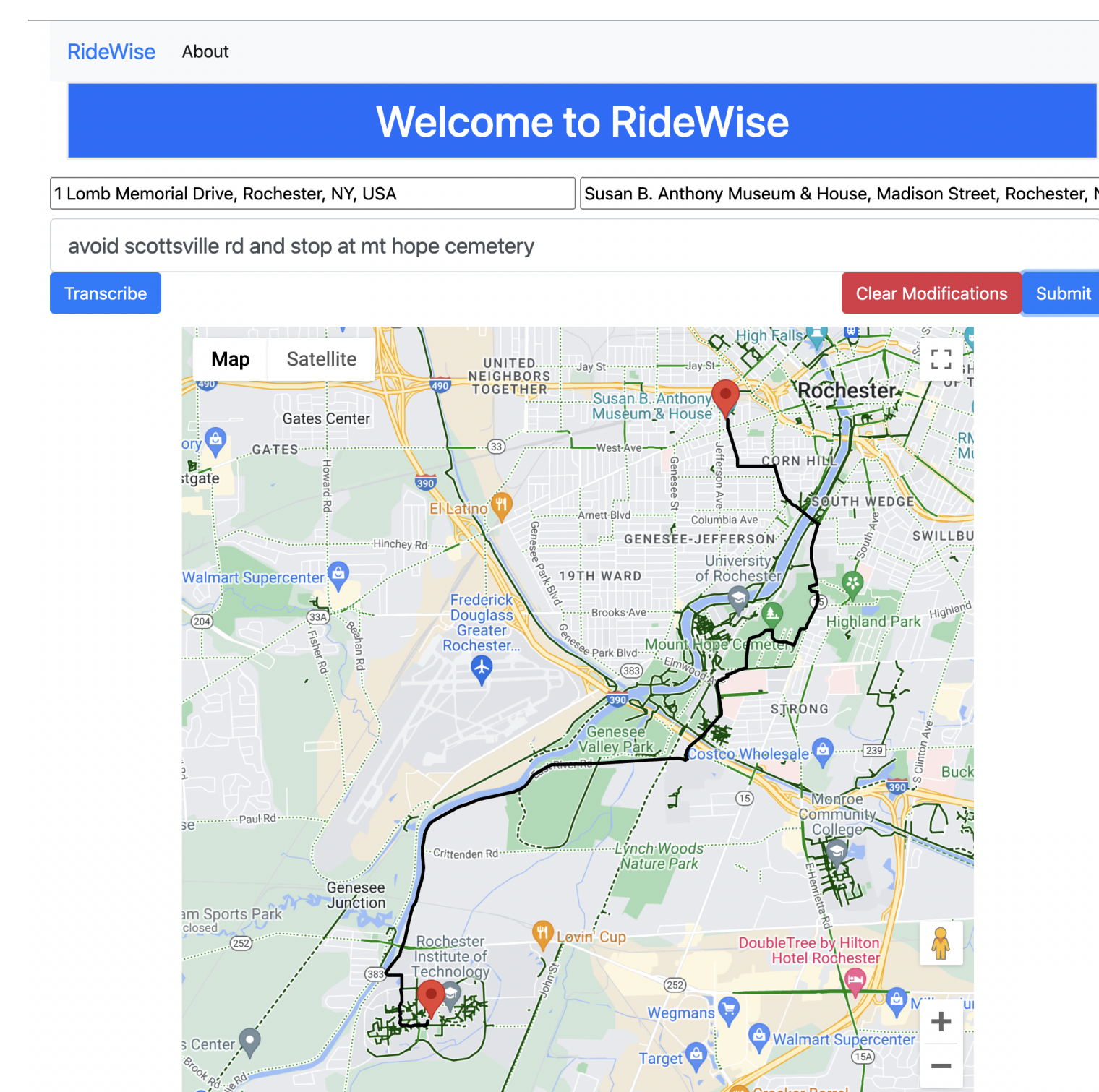


Figure 1: Application interface with example route and modifications.

System Architecture

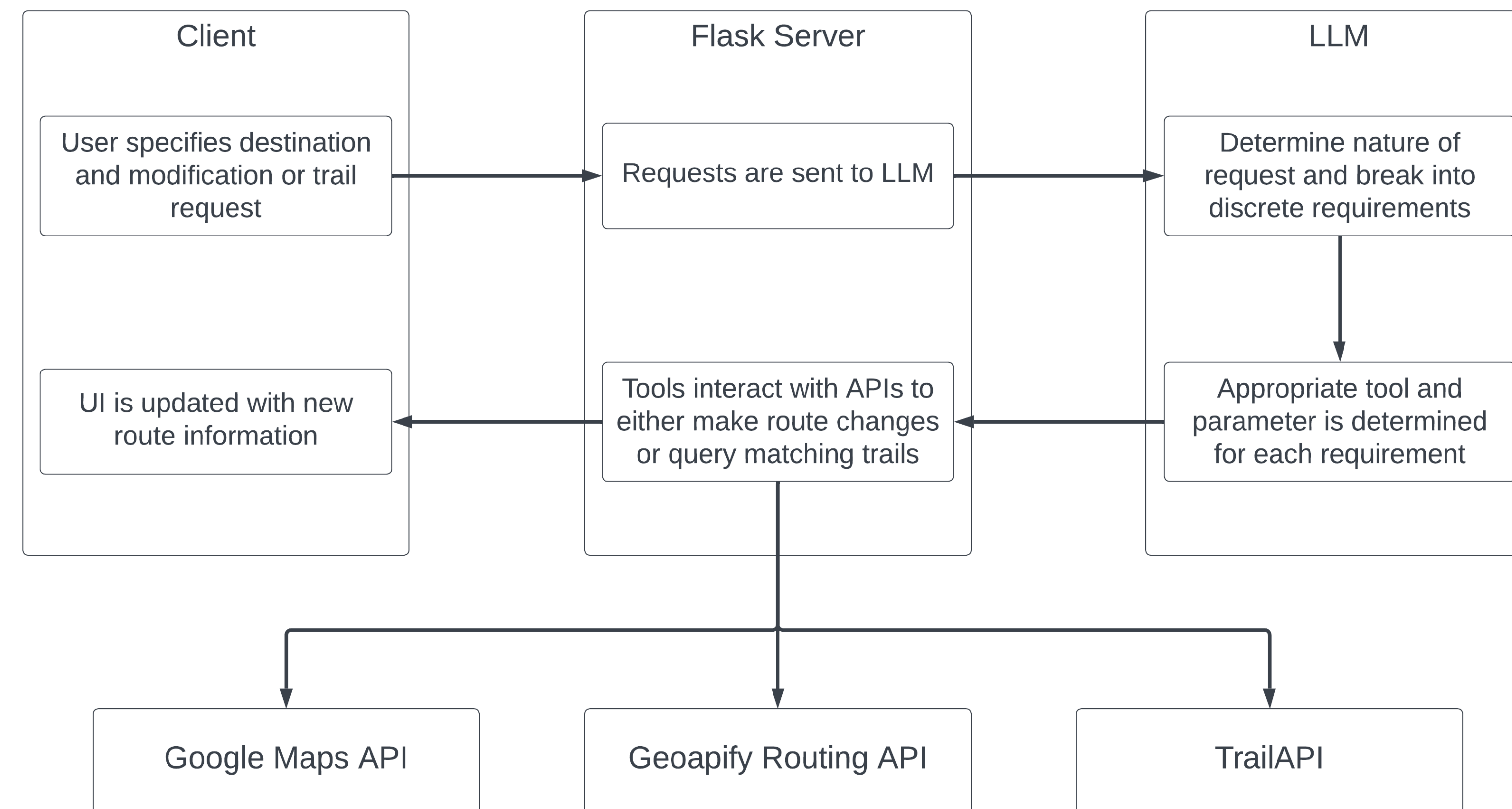


Figure 2: Simplified system architecture diagram demonstrating the connections between the primary components: client, server, and LLM.

Client

- Handles user interaction and displays generated routes and trails. Speech-to-text capability is included for hands-free modification.

Server

- Routes communication between client, LLM, and APIs.
- Employs extensive error-checking logic to safeguard against LLM errors and ensure user requests are reflected in service.

LLM

- **Modification or Trail?:** Determines if request is related to a route modification ① or an informational request about local trails ②.
- **Splitting Input:** Splits input into discrete requests.
- **① Choosing Change:** Decides which type of change is being requested as part of each request.
- **② Extracting Trail Information:** Extracts relevant details about trail information request.

Results

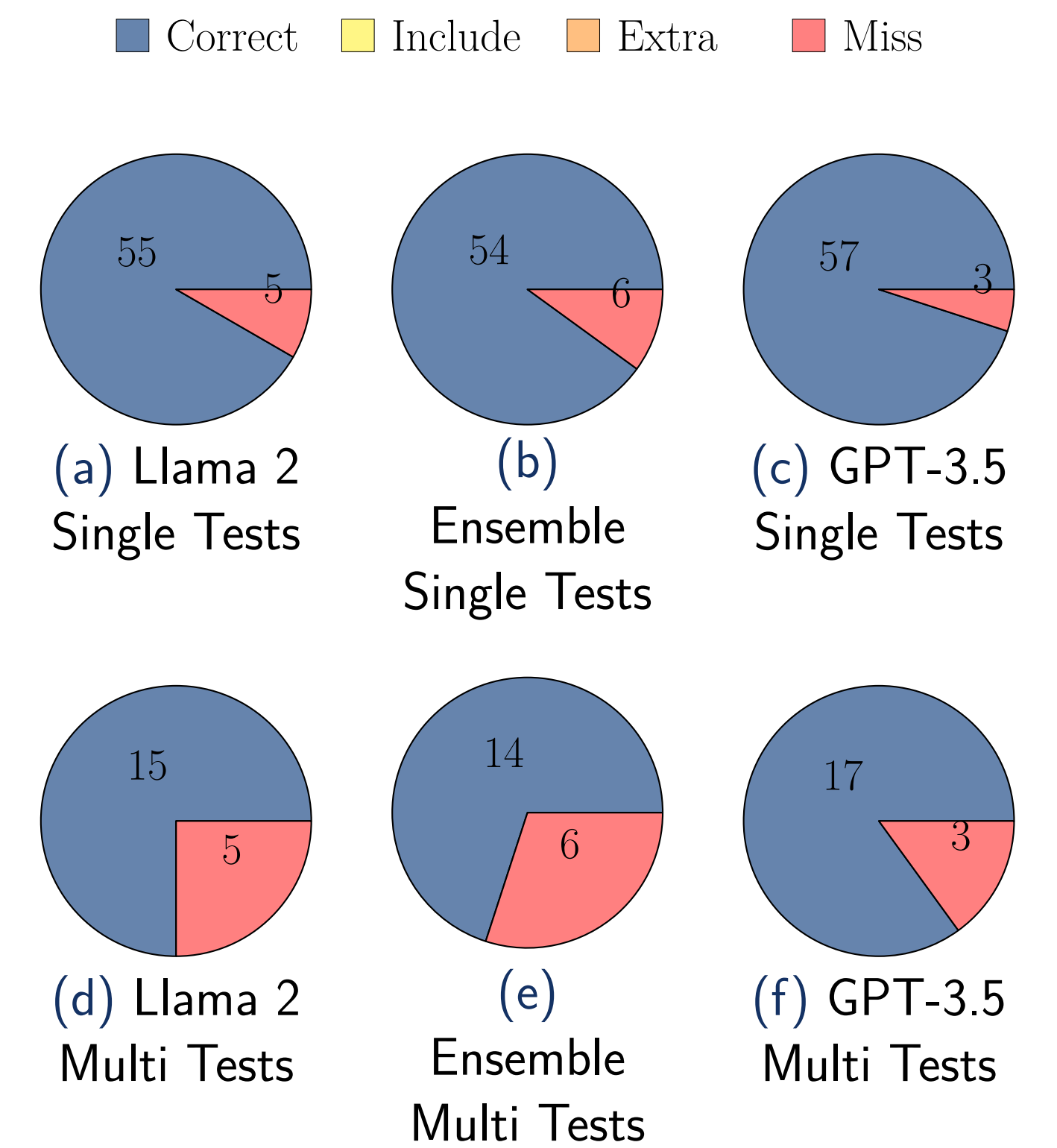


Figure 3: Did route match requests?

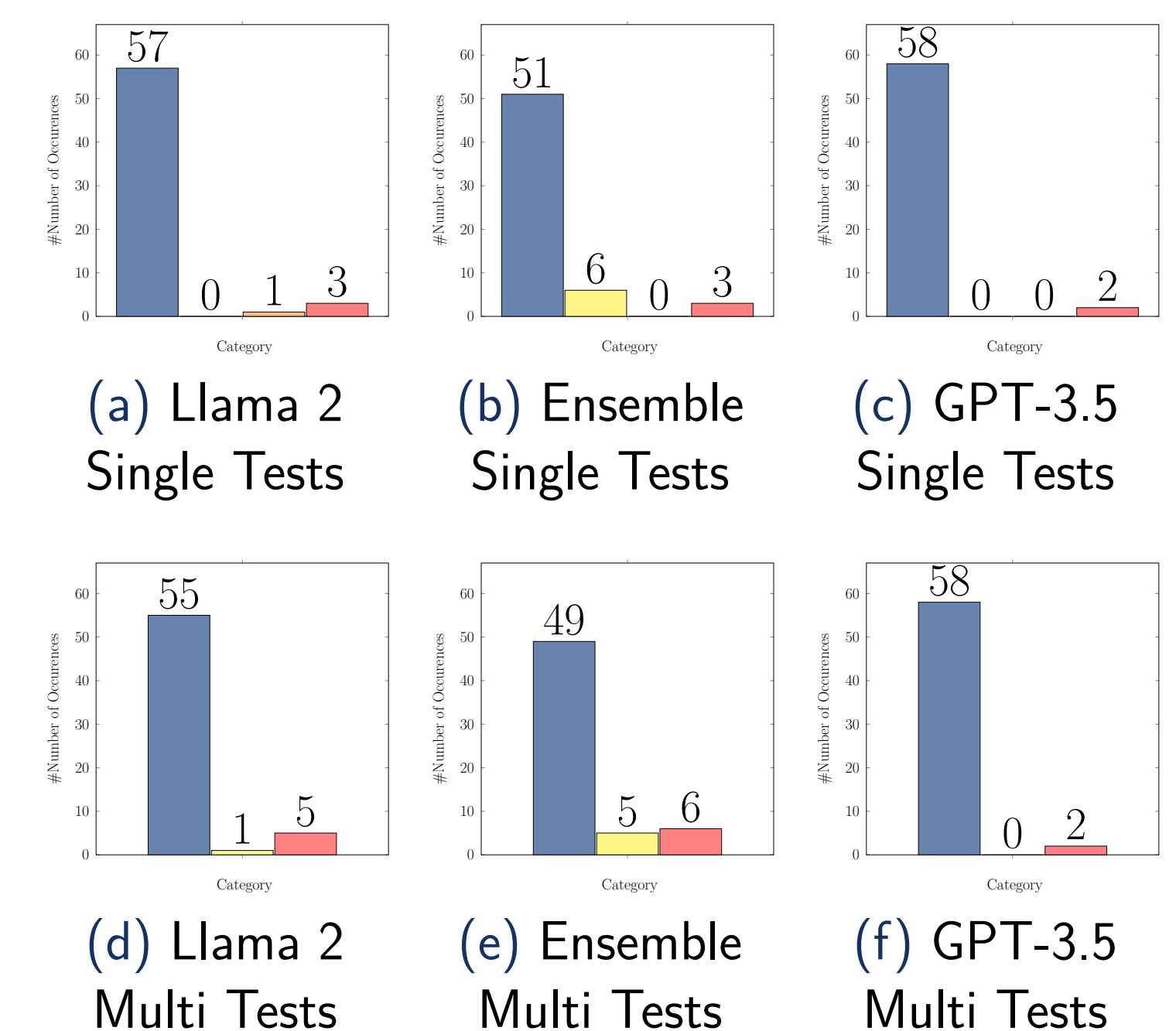


Figure 4: Did LLM parse requests correctly?