UNIVERSITÀ DEGLI STUDI DELL'AQUILA

Software Architectures

Large Language Models for User Goals

Author :
Ruben Huygens
Juliette Waltregny
Samuele Ramassone
Anusha Annengala

Professor : Henry Muccini
Year : 2023-2024

Large Language Models for User Goals

April 4, 2024

Contents

1 Introduction

In this project, we were asked to improve the user experience during the Street Science days in L'Aquila. The goal is to expand the capabilities of the NdR Mobile App, specifically its architecture, to incorporate a chatbot enabling the user to request one or multiple services from the app incorporating essential components to enable the utilization of Large Language Models (LLMs). These LLMs serve as a mechanism to derive user goals from a given natural language text of a users. Using LLMs, we identify user intentions and parse them into appropriate grammatical structures, and further make use of goal modeling mechanisms to identify the functionalities required to accomplish the user goal. The goal of the Ndr application is to allow users to access events, locate parking spaces and easily navigate the city, as well as enable better crowd management and improve the quality of the visit experience using its microservices-based architecture. Our component serves as an extension of the current application, which focuses on identifying user intentions and using them to change the application's behavior adaptively. By incorporating user objectives identified through natural language models, we aim to refine the capabilities of the application and optimize the visitor experience by allowing them to access the services offered during the Street Science days in an easier and more intuitive way, through use of a UI to exchange messages with the language model.

2 Project Planning

Clear distribution of roles and responsibilities is crucial for effective project management and ensuring the timely completion of tasks. We have outlined the roles, tasks and responsibilities for each team member involved in the project.

2.1 Team members

All the team members worked on development and implementation of the project. In addition, we had to assign more specific roles to ensure that the project management was as smooth as possible, the roles being 'Leader' and 'Communication Strategist'.

Ruben Huygens

1. Role: **Leader of the project and Developer**. Oversees the entire project, including planning, execution, and monitoring. Participates in the technical implementations of the project.

2. Tasks: Develops project plan, assigns tasks to team members, writes code, develops features, and integrates functionalities.

3. Responsibilities: Communicates project updates and future steps, decides when meetings should be held. Resolves conflicts and obstacles hindering project progress. Makes sure the project meets technical aspects and requirements.

Juliette Waltregny

1. Role: **Communication strategist and Developer**. Ensures effective dissemination of project information to stakeholders and between team members. Participates in the technical implementations of the project.

2. Tasks: Crafts messaging, manages communication channels, organizes meetings when the Leader has decided so and makes sure the meeting information is known by every member. Writes code, develops features, and integrates functionalities.

3. Responsibilities: Develop a comprehensive communication plan outlining key messages, target audience, and communication channels. Makes sure the project meets technical aspects and requirements.

Samuele Ramassone

1. Role: **Developer**, Implements the technical aspects of the project.

2. Tasks: Writes code, develops features, and integrates functionalities.

3. Responsibilities: Ensures the project meets technical requirements and standards, tests and debugs code to identify and fix any issues.

Anusha Annengala

1. Role: **Developer**, Implements the technical aspects of the project.
2. Tasks: Writes code, develops features, and integrates functionality.
3. Responsibilities: Ensures the project meets technical requirements and standards, tests and debugs code to identify and fix any issues.

2.2 Development responsibilities

In particular, we decided on the technical responsibilities of the different members from the development side as the following:

- **Identification of Functions**: this responsibility was attributed to *Samuele*, he was therefore in charge of prompt engineering and teaching the LLM to output a list of the functions identified from the user's input. The implementation of this phase is detailed in Section 5.
- **Identification of Logical Operators Phase and Conversion to Grammar**: this phase consists of identifying the underlying relationships between the intents of the user and correctly mapping them to the already identified functions. We also know that the NdR system can only understand a user goal in the correct grammar, we therefore had to make sure it was correctly formatted. This responsibility was attributed to *Anusha* and *Juliette*. The implementation of the phase is detailed in Section 5.
- **Interaction between LLM Components and User Interface**: this task was done in parallel of the two previous one. It makes sure the interaction between the different LLM components and the user interface is correctly articulated. This responsibility was attributed to *Ruben*. The implementation of the phase is detailed in Section 5.

3 Understanding the context and objectives

The primary objectives of this phase were to familiarize ourselves with the project's context, study relevant articles, and gain insight into the already existing system. In addition, we took some time to understand the behaviour of the LLM and how to interact with it thorough reading of the documentation.

Objectives

- **Study Articles**: The team was tasked with reading provided articles relevant to the project's domain. These articles served as foundational knowledge, providing insights into the current state of research, challenges, and advancements in the field.
- **Study LLM Functioning**: Understanding the functioning of the LLM was essential for tailoring our approach to the project. We aimed to analyze its capabilities, limitations, and potential areas for improvement

Key findings and observation

- **Article Analysis**: The articles provided valuable insights into the requirements and approaches to this project. This understanding helped us identify potential gaps or areas where our project could contribute.
- **LLM Functioning**: Through the study of LLM functioning, we gained a deeper understanding of its architecture, training data, and performance metrics.

4 Design

In this section we will explain our design implementations. To do so, we have outlined the design phase of the project, focusing on identifying architectural significant requirements, selecting technologies, and LLMs. Additionally, we have highlighted the design decisions made during this phase and the agreement reached with stakeholders on the proposed design.

Choice of Technologies and LLMs

LLM decision

We based our decisions following different criteria:

- **Token limit**: Tokens are the basic unit of text than an LLM can process (as input and output). Here are the respective token limits:
  1. LLama 2: 4096

    2. GPT 3.5: 4096 (turbo reaches 16 384)
    3. GPT 4: 8192 (most powerful one reaches 32 768)

- **Speed and efficiency**: LLaMA 2 stands out for its impressive speed and efficient use of resources, surpassing GPT-4 in these aspects. This feature allows LLaMA 2 to maintain a good balance between being fast and accurate. On the other hand, GPT-4, with its extensive range of features, tends to require more computational power, which can make it slower in comparison.

- **Accuracy**: In a recent study by Anyscale, they discovered that Meta's LLaMA 2 70B is nearly as accurate as GPT-4 when it comes to summarizing information correctly. However, GPT-4 still leads in more specialized tasks, such as the HumanEval coding benchmark.

- **Price**:

    1. LLama 2: Open-source, no direct cost for the model itself.
    2. GPT 3.5: Commercially licensed, costs based on usage volume and API access.
    3. GPT 4: Expected to follow a similar commercial licensing model as GPT-3.5, potentially with higher costs due to enhanced capabilities

.

UI Framework decision

We have considered two UI frameworks, i.e., React and Flutter and differentiated them based on the following categories:

- **Programming language**:

    1. **React Native**: Utilises JavaScript, a widely-used and well-established language. Its large talent pool and extensive libraries contribute to faster development cycles and easier maintenance.
    2. **Flutter**: Relies on Dart, a language developed by Google specifically for Flutter. While less widespread than JavaScript, Dart is easy to learn and offers a reactive and streamlined development experience.

- **Community size**: Both Flutter and React Native have many contributors and an active community. That makes it easier for beginners like us to develop solutions to our problems. Since React Native is backed by Facebook, the framework has a lot more resources at its disposal than Flutter, which means we'll find answers much faster should we hit any problem while working on Flutter.

- **Development Flexibility**: React Native allows for a more flexible development approach, enabling developers to use native modules when necessary. This can be advantageous for integrating device-specific features or addressing performance-critical aspects of your chatbot UI.

- **Learning Curve**:

  1. **React Native**: Easier for developers already familiar with JavaScript and React.
  2. **Flutter**: Requires learning Dart, but is straightforward for developers with experience in object-oriented languages. The framework itself is easy to pick up

  .

5 Implementation

We have outlined the implementation phase of the project, detailing the incremental steps taken to achieve project objectives. It also summarizes the outcomes and lessons learned during the implementation process.

Implementation stages

The implementation phase was executed through incremental steps to ensure steady progress and timely feedback. Key steps included:

- Setup and Environment Configuration: Establishing development environments and configuring necessary tools and frameworks.

- Basic Functionality Implementation: Implementing basic functionalities according to the design specifications, such as creating prompts and templates.

- Iterative Development and Testing: Following an iterative development approach, continuously enhancing functionalities and conducting thorough testing to identify and address bugs and issues.

- Integration and System Testing: Integrating individual components and conducting system-level testing to ensure seamless interaction and overall system functionality.

- User Interface Refinement: Refining the user interface and ensuring a seamless user experience.

Setup and Environment Configuration

- We used React framework for the user interface of the chatbot which easily allows users to input a natural language text.

- We used Llama model to identify the user goals from the natural language text of the user, specifically the ”Llama-2-13B-chat-GGUF” version of the model in our system

- We employed Docker, a containerization platform, to facilitate the execution of our model within our local computing environment.

Basic Functionality Implementation

User Interface:

- Description: The user interface (UI) serves as the primary interaction point between users and the system. It encompasses the design, layout, and functionality of the graphical interface through which users interact with the application.

- Implementation Details: We used React framework to develop a simple chatbot user interface where the user can easily input a natural language text. The chatbot then calls the API to fetch the identified user goals and displays it to the end user.

LLM Model:

- Description: The LLM (Large Language Model) serves as the core component of the system, responsible for identifying goals based on given prompts and context. It leverages advanced natural language processing techniques to understand the input.
- Implementation Details:
  1. The two main tasks here were to identify the functions from the natural language text of the user and to identify the relationship between these functions.
  2. To identify the functions, we considered the natural language text of the user as input and passed a prompt to the model to identify the functions [weather checking, ticket availability, event booking, parking recommendations] in context of the user input.
  3. To determine the relationship between the identified functions, we consider the user input and the identified functions as the input. Both of them are passed to a notebook for ease of processing which is then passed to the model along with the prompt to identify the appropriate relationships.
  4. We also added features where the user can verify if the identified functions and relationships are accurate.If not, the model is called again to determine the right functions and relationships.
- Challenges and Solutions: We faced issued while running the model on a docker container because our computers do not have enough computational resources to run the model. As a solution, we used a MacBook Pro computer to continue testing the model.

Grammar generation:

- Description: Grammar generation involves the generation of syntactically correct and contextually relevant grammar by the system in accordance with MiLA4U specifications.
- Implementation details: We parsed the output of the LLM model to convert into a format of the grammar specified in MiLA4U

Iterative Development and Testing

- Description: Iterative development is an approach where the software development process is divided into small, manageable iterations or cycles. Each iteration typically involves the implementation of a subset of features, followed by testing and evaluation.

- Define Requirements: Clear and detailed requirements for the chatbot, including its functionality, user interface design, and integration with other systems.

- Initial Design and Prototyping: Initial design and prototype of the chatbot to visualize its user interface and basic functionality. This prototype can be used to gather feedback and identify areas for improvement.

- Develop Core Functionality: We implemented the core functionality of the chatbot, focusing on key features such as identifying functions and relationships, generating accurate grammar and integration with the user interface.

- Incremental Feature Development: Incremental approach to feature development, prioritizing the most important and high-impact features first.

- User Feedback and Iteration: Use the feedback to inform future iterations of the chatbot, focusing on addressing user pain points and enhancing usability.

- Monitoring and Performance Optimization: Chatbot's performance, including response times, error rates, and user engagement metrics.

- Testing: Through testing to validate the functionality of individual components within the chatbot.

6 Conclusion

We successfully created a system to identify the user goals from the natural language text of the user. The design decisions have been explained in detail along with the implementation of the project. The roles and responsibilities of the team members are well defined which made an efficient working environment and a successful project possible. In conclusion, our system generates the right grammar specified in MiLA4U based on the natural language input of the user.

7 References

1. Hugo Touvron, Thibaut Lavril, LLaMA: Open and Efficient Foundation Language Models, Feb 2023.

2. Anjali Khurana, Hari Subramonyam, Parmit K Chilana, Why and When LLM-Based Assistants Can Go Wrong: Investigating the Effectiveness of Prompt-Based Interactions for Software Help-Seeking, Feb 2024.

3. Leuson Da Silva, Jordan Samhi, Foutse Khomh, ChatGPT vs LLaMA: Impact, Reliability, and Challenges in Stack Overflow Discussions, Feb 2024

4. Prateek Rawat, Archana N. Mahajan, ReactJS: A Modern Web Development Framework, Nov 2020

5. Rishi Vyas, Comparative Analysis on Front-End Frameworks for Web Applications, July 2022