HackAI PS Round 2

Team ID: Hack-230494

Techfest IIT Bombay 2023

Automated Customer Support Agent

Using Natural Language Processing for Customer Queries

TECHNICAL REPORT

Repo Link :Github

1 Problem Statement

The objective of this project was to develop an automated customer support system utilizing natural language processing techniques. The project aimed to address customer inquiries for a hypothetical company, Hitachi, by creating agents capable of understanding and responding to queries, escalating issues when necessary, and improving over time through learning from interactions.

2 Project Overview

Our team created a command line application using the uagents library in Python. Two distinct agents were developed: the 'generator' and the 'answerer.' The 'generator' agent is responsible for creating potential responses to customer queries, while the 'answerer' utilizes Google's RAG and PALM APIs to select the best response based on a document of information generated for Hitachi. These agents employ natural language processing techniques to understand user queries, provide relevant responses, and improve their performance over time through interactions.

3 Solution Approach

To address the problem statement, we employed the RAG (Retrieval Augmented Generation) and PALM API provided by Google to enhance the response generation process. By leveraging these APIs and following the methodology outlined in the GenRead research paper, we implemented a multi-step approach. First, the 'generator' agent creates potential responses to a query. Subsequently, the RAG algorithm is applied to these responses to determine the optimal output based on the Hitachi information document.

4 Folder Structure

```
.
 src/
    agents/
       generator/
          generator.py
       answerer/
           answerer.py
    messages/
       general.py
    utils/
       initiate.py
    document.txt
    main.py
 .gitignore
 README.md
 requirements.txt
```

5 Implementation Details

The agents interact with the uagents library, which provides agent-based systems. Additionally, the RAG and integration with PALM APIs allowed seamless communication for response generation. The 'generator' agent uses techniques suggested by the GenRead paper to generate potential responses, while the 'answerer' applies RAG over these responses to select the most suitable answer for a given query.

Generator Agent Implementation The generator agent is responsible for initiating conversations, generating potential answers to user queries, and interacting with the answerer agent for further refinement.

Answerer Agent Implementation The answerer agent receives input from the generator agent, refines the responses generated by leveraging a language model, and provides enhanced answers to user queries.

We also keep on updating the document for retrieval based on the question and answers so that the agent can keep on learn and improve over time.

6 GenRead Research Paper

The GenRead research paper suggests a methodology for response generation by creating potential responses and employing RAG to obtain the best output. Our project aligned with this approach, using the 'generator' to generate diverse responses and then applying RAG for optimal response selection. This method significantly improved the accuracy and relevance of responses provided by our agents.

7 Results

Throughout the project, we observed significant improvements in the agents' ability to handle customer inquiries. Metrics such as response accuracy, speed, and user satisfaction demonstrated noticeable enhancements over the development phases. Challenges were encountered in RAG selection process and optimizing response generation, but these were mitigated through iterative improvements.

8 Conclusion

In conclusion, the project successfully developed an automated customer support system using the uagents library and Retreival Augmented Generation(RAG) and Google's PALM APIs. The agents effectively handled customer queries, escalated issues when necessary, and improved over time through learning from interactions. Future enhancements could involve refining the response generation process and exploring additional natural language processing techniques.

9 References

1. GenRead Research Paper: GenRead Paper
2. uagents Python Library Documentation: uagents Documentation
3. PALM API Documentation: Google API Documentation