

Intelligent Chatbot for Amazon Reviews Analysis

Daniele Mingolla

Problem Statement

Our client, an established fashion brand, is entering Amazon's marketplace and is concerned about understanding customer experiences through online reviews. They need a solution that can analyse Amazon reviews to identify key customer sentiments, themes, and factors driving positive reviews. This information is essential for shaping product strategies and improving customer satisfaction. The challenge is to efficiently extract meaningful insights from a large volume of reviews and provide a user-friendly tool for the client's team to navigate and understand customer sentiments.

Approach and Solution

In response to the client's requirement for a comprehensive analysis of Amazon reviews to enhance customer experiences, we are adopting a strategic approach that encompasses advanced natural language processing techniques and an innovative web application. Our solution leverages LangChain for enhanced language processing capabilities and Streamlit for the creation of an intuitive web-based chatbot. Additionally, we integrate OpenAI's API to further enrich the chatbot's conversational abilities.

We will utilise **LangChain**, an open-source library designed to empower developers in building applications fueled by large language models (LLMs). LangChain provides us with an orchestration framework, allowing seamless chaining of various prompts. With LangChain, we can efficiently process Amazon reviews, extract sentiments, identify recurring themes, and derive meaningful insights from the customer feedback.

Our approach involves the development of an interactive web application using **Streamlit**, a versatile Python framework. This web app will serve as an interface for the client's team to interact with the processed Amazon reviews data. The intuitive nature of Streamlit will enable users to explore reviews and delve into specific topics with ease.

To enhance the chatbot's conversational capabilities, we will integrate **OpenAI's API**. This integration enables the chatbot to engage in more contextually relevant and dynamic conversations, resulting in a more natural and fluid interaction with users.

Chatbot Features

Smart Chatbot for Amazon Product Reviews

This chatbot offers a natural conversational experience, helping users explore reviewer insights and learn about good reviews. Remember to upload your JSON files from the sidebar before asking questions!



Hi! Based on the reviews I sent you, what constitutes a good product?



Based on the reviews provided, a good product is one that is described as "great" or "excellent" in terms of quality and value. It should meet the expectations of the customer and be comfortable to use.



What's the reasoning behind this conclusion?



The reasoning behind the conclusion that a good product is one that is described as "great" or "excellent" in terms of quality and value, meets the expectations of the customer, and is comfortable to use is based on the positive feedback and satisfaction expressed by the customers in the given context. The customers mention that the product is comfortable, consistent, and meets their expectations. They also mention that the product is accurately described and the photos provided are accurate. These positive comments indicate that the product is of good quality, provides value for money, and meets the needs and preferences of the customers.

Enter your question...



User-Friendly Interface

The chatbot is presented through a user-friendly web interface powered by Streamlit. Users can easily input their questions and receive informative responses in a conversational format.

Contextual Memory

The chatbot is equipped with an intelligent memory system that retains the context of ongoing conversations. This memory allows the chatbot to recall previous interactions and responses, enabling coherent and relevant discussions.

Document Upload and Processing

Users can easily upload JSON files containing Amazon product reviews. The chatbot's user-friendly design ensures that users understand the requirement for consistent file structure across the uploaded documents.

Efficient Data Preprocessing

Once the user uploads JSON documents and initiates processing, the application efficiently extracts the relevant columns ("overall" and "reviewText") from the documents. This preprocessing step ensures that only essential information is considered for the subsequent analysis. The chatbot intelligently splits the textual content into smaller, manageable chunks using a character-based splitter. This approach enhances the chatbot's ability to process and understand larger textual inputs, contributing to a seamless user experience.

Modular and Extensible Code

Our solution is built with modularity in mind, facilitating easy expansion and customization. Components such as chat models, memory tracking, and data processing are modular, ensuring flexibility and maintainability. It's easy to change the vector store, the LLM model or the embedder if needed.

Effortless Containerization with Docker

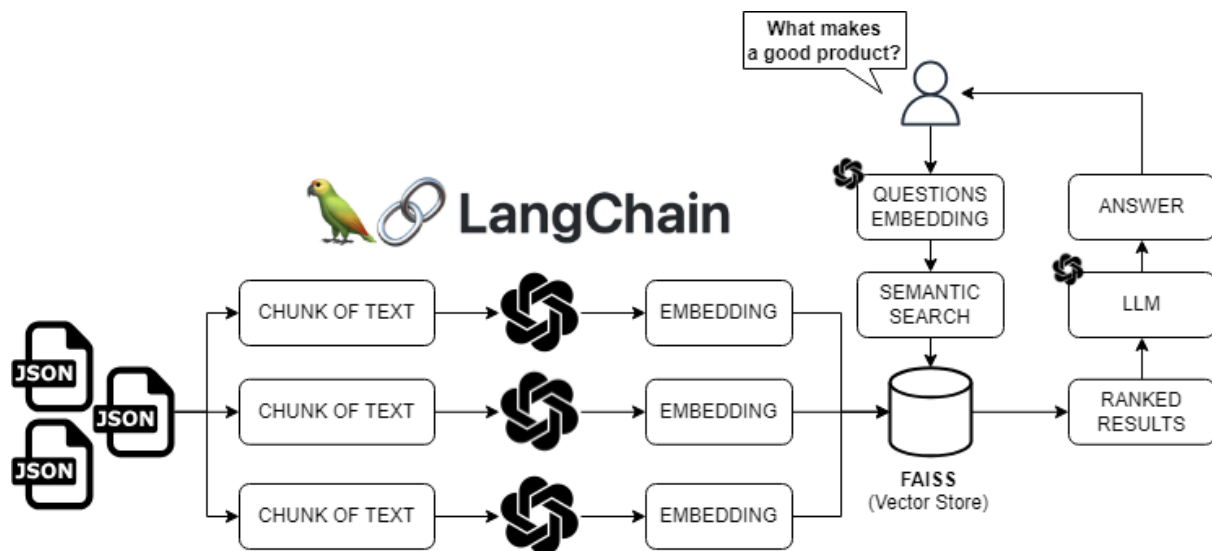
The entire chatbot system, including its components, can be containerized using Docker. This simplifies deployment and ensures consistent performance across various environments.

Example of questions to ask

Here's a list of example questions that users can ask the chatbot to explore Amazon reviews:

- "Can you summarize the overall sentiment of the reviews for product X?"
- "What are customers saying about the quality of product Y?"
- "Are there any common complaints in the reviews for brand Z?"
- "Tell me about the positive features of product A according to the reviews."
- "Compare the customer satisfaction levels for product B and product C."
- "Which product has the highest ratings among those reviewed?"

Architecture



1. **Text Splitting:** This initial step involves segmenting the input text into smaller, more manageable portions. Employing a character-based text splitter, the text is divided into coherent units, facilitating better processing.

2. **Embedding with OpenAI:** Once split, the text chunks undergo embedding using OpenAI's embedding techniques. These embeddings encapsulate the underlying semantic essence, converting the textual content into numerical representations that encapsulate both context and meaning.

3. **Vector Store Creation:** The embedded text chunks are aggregated and housed within a vector store. This data structure is meticulously designed for proficient similarity searches. Making use of the FAISS library, which specialises in similarity search and clustering, the vector store ensures optimised search operations.

4. **User Input and OpenAI Embedding:** When a user poses a question, the input is seamlessly embedded using OpenAI's embedding approach. This transformation prepares the user's query for further processing and analysis.

5. **Semantic Search and Matching:** The newly embedded user query is evaluated against the embedded text chunks previously stored in the vector store. A semantic search and matching process ensues, identifying the text chunks that most closely correspond to the user's query based on their embedded representations.

Setup

1. Install Docker on your system if not already installed.
2. Unzip the `code_docker.zip` file.
3. Open a terminal and navigate to the root folder of the project.

4. Create a `.env` file within the chatbot directory and insert this line in it:
`OPENAI_API_KEY = <your_api>`.
5. Execute the command `docker-compose up`.
6. Access the chatbot in your web browser at `http://localhost:8501`.

Future Enhancements

Interactive Visualizations

Extend the chatbot's capabilities to provide interactive visualizations. Users could request graphical representations of review sentiments, trends, and other relevant insights.

Multi-Modal Integration

Explore the integration of multi-modal capabilities, such as analyzing images or video content associated with reviews. This could add a new dimension to the chatbot's analysis and responses.

Integration with Additional Data Sources

Integrate data from other sources, such as social media or customer service inquiries, to provide a more comprehensive understanding of customer experiences.