|  |  |
| --- | --- |
|  | Proprietary  Information |

Orbit Research website Chatbot

Software Requirement Specification

12th June 2023

Version 0.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Date | Description of Changes | Author |
| 0.0 | 28th July , 2023 | Created | Dhruv |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Introduction

This Software Requirement Specification (SRS) document describes the features, functionality, and design constraints of a Document Chatbot application. The application is designed to ingest a collection of documents, break them down into manageable chunks, index these chunks using the Pinecone vector database, and then provide conversational capabilities using OpenAI's GPT-3.5-turbo model (or GPT-4, if available). The chatbot will be deployed on the OrbitResearch website.

System overview

The system consists of two main components:

Document Ingestion and Indexing: This phase involves breaking down documents into chunks of text, embedding these chunks using OpenAI's language model, and storing the resulting vectors in Pinecone, a vector database.

1. Chatbot Interface: This phase involves deploying a chatbot on the OrbitResearch website that can retrieve context-relevant information from the indexed document chunks and generate responses to user queries.
2. The Document Ingestion and Indexing component uses Node.js and the Pinecone and OpenAI APIs. The Chatbot Interface uses the OpenAI API for generating responses and a suitable web technology (such as Node.js with Express) for deployment on the website.

Operating environment

The software is designed to run in a Node.js environment. The Pinecone and OpenAI APIs are required, and the system must have access to the internet to interact with these APIs. The system should also have access to the document folder from which documents are ingested. The chatbot will be deployed on the OrbitResearch website, which requires a suitable web server environment.

Functional Requirements (Responsibilities)

1. The software shall provide a command (e.g., "npm run ingest") to start the document ingestion and indexing process.
2. The software shall read documents from a specified document folder.
3. The software shall break down documents into chunks of text using the Langchain library.
4. The software shall generate embeddings for the chunks of text using the OpenAI API.
5. The software shall store the embeddings in a Pinecone vector database with an index of 1536 dimensions, using cosine similarity for vector comparison.
6. The software shall provide a chatbot interface on the OrbitResearch website.
7. The software shall use the OpenAI API and the indexed document chunks to generate responses to user queries.
8. The software shall ensure that the chatbot's responses are relevant to the provided context.
9. The software shall handle errors gracefully, providing meaningful error messages to the user or system administrator.

Users (Interface) requirements

The system should interact with the following entities:

Users:

* The user should be able to interact with a chatbot interface.
* The user should be able to ask questions relevant to the ingested documents.

Input:

* User queries in the chatbot interface.
* Documents for ingestion placed in the specified document folder.

Output:

Responses generated by the chatbot based on the user's queries and the context provided by the ingested documents.

Use cases

**Use Case 1:** Document Ingestion and Indexing

* The system administrator runs the document ingestion and indexing command.
* The software reads the documents, breaks them down into chunks, generates embeddings, and stores these embeddings in the Pinecone vector database.

**Use Case 2:** Chatbot Interaction

* The user enters a query in the chatbot interface on the OrbitResearch website.
* The software retrieves relevant context from the indexed document chunks and generates a response using the OpenAI API.
* The response is displayed in the chatbot interface.

Database requirements

The application uses Pinecone, a vector database, to store the embeddings of the document chunks. The database has an index of 1536 dimensions, and cosine similarity is used for vector comparison.

Testability requirements

The software should include tests for the document ingestion and indexing process, as well as for the chatbot response generation. These tests should ensure that the software correctly processes and indexes documents and generates appropriate responses to user queries.

Logging requirements

All significant events during the document ingestion and indexing process should be logged, including the start and end of the process, any errors or exceptions that occur, and any issues with reading documents or interacting with the Pinecone or OpenAI APIs.

The chatbot should log all user queries and the generated responses. Any errors or exceptions that occur during the response generation should also be logged.

Manufacturing requirement

No specific manufacturing requirements are needed for this extension.

Budget

Pinecone’s starter pack is free. OpenAI text embedding API one time cost to embed data in Pinecone vector DB. OpenAI model 3.5 turbo usage cost.

Human Resource requirement

Developers, QA testing

Hardware Set ups requirements

A computer with Node.js installed is required. An internet connection is necessary for interacting with Pinecone and OpenAI APIs. The OrbitResearch website must have the capability to host the chatbot interface.

Non-functional requirements

The application must be developed using Node.js.

The application should handle errors gracefully, providing clear error messages.

The application should be compatible with a variety of document formats for ingestion.

The chatbot interface should be user-friendly and intuitive.

The chatbot should provide responses within a reasonable time frame.