Website Analysis and OpenAI Integration

Project Overview

This project focuses on extracting and summarizing content from websites using Python libraries such as `requests` and `BeautifulSoup`. Additionally, the project integrates OpenAI's GPT model to generate meaningful summaries of extracted content.

Data Extraction and Processing

1. **Fetching Website Content:**
    - The `requests` library is used to send HTTP GET requests to retrieve webpage content.
    - The `User-Agent` header is set to mimic a browser request and avoid bot detection.
2. **Parsing and Cleaning HTML:**
    - The `BeautifulSoup` library is used to parse HTML content.
    - Irrelevant elements such as `<script>`, `<style>`, `<img>`, and `<input>` tags are removed to extract meaningful text.
3. **Extracting Relevant Text:**
    - The webpage title is extracted.
    - The cleaned text is formatted for further processing.

OpenAI Integration

1. **Connecting to OpenAI API:**
    - Environment variables are loaded using `dotenv`.
    - The OpenAI API key is retrieved and validated before making API calls.
2. **Generating Summaries:**
    - The extracted text is processed using OpenAI's `gpt-4o-mini` model.
    - A system prompt is provided to guide the model in generating structured and concise summaries.
3. **Handling API Responses:**
    - Responses are parsed to extract the generated summary.
    - The summary is displayed in Markdown format for better readability in Jupyter Notebook.

Implementation Details

1. **Class Definition:**
    - A `Website` class is created to encapsulate URL fetching, HTML parsing, and text extraction.
    - The class retrieves the title and relevant text from the webpage.
2. **Prompt Engineering:**
    - A predefined system prompt ensures the summary is focused and excludes navigation-related content.

    - A function constructs a user prompt dynamically based on the extracted content.
3. **Function Calls:**
    - `messages_for(website)`: Prepares messages for OpenAI API calls.
    - `summarize(url)`: Fetches and processes website content, then generates a summary.
    - `display_summary(url)`: Formats and displays the generated summary in Markdown.

Example Execution

Website Analyzed: Quantum Computer - Simple English Wikipedia

Extracted Summary: A quantum computer is a type of computer that leverages principles from quantum mechanics, such as superposition and entanglement, to process data in ways that classical computers cannot. Unlike traditional computers that store information in binary (0s and 1s), quantum computers use qubits, which can exist in multiple states simultaneously. This allows them to perform certain calculations exponentially faster than classical systems. Although still in early development, quantum computing has potential applications in cryptography, optimization problems, and complex simulations.

Conclusion

This project demonstrates how to extract and summarize website content effectively using a combination of web scraping techniques and AI-based language models. The integration of OpenAI's GPT model enhances the ability to generate insightful and structured summaries from raw webpage text.