AI Search Assistant for Complex Websites

Introduction:

Government websites are often large and complex, with vast amounts of content that can overwhelm users. The challenge of navigating these websites to find specific information can be time-consuming and frustrating. While these sites typically include search features, they often fail to deliver accurate results, especially when users are looking for information within documents like PDFs or DOCX files. Furthermore, the structure of these websites can be difficult to navigate, making it hard for users to locate the specific content they need.

To address these challenges, we developed a custom AI search assistant tailored to government websites. The goal was to provide users with an intelligent assistant that could help them quickly find relevant information. By training the chatbot on content specific to each website, it became better equipped to understand the context and provide accurate, relevant responses to user queries. This enhanced the chatbot's ability to assist users and helped streamline the process of finding information.

The chatbot also organizes content into categories, allowing users to narrow their search and avoid feeling overwhelmed by the sheer volume of data. By maintaining context of a conversation, the chatbot ensures a smooth and continuous interaction, making the user-experience more natural and efficient. This approach makes it easier for users to find exactly what they need, improving the overall navigation and functionality of the website.

Overall, the custom AI search assistant significantly enhances the user experience on government websites. It simplifies the process of searching for information, reduces frustration, and provides a more seamless and efficient way to navigate through complex and extensive content.

Client Details:

Name: Confidential | **Industry:** Software | **Location:** USA

Technologies:

ReactJS, Python FastAPI, MariaDB , LangChain, GPT model, Quadrant vector database, Google Analytics, Oracle VM

AI Search Assistant for Complex Websites

Project Description:

The project was mainly categorized into two different phases:

1. **Chatbot Training**

   Training the chatbot for government websites involves tailoring the AI model to understand the unique content and structure of that specific site. This process includes feeding the chatbot with relevant data from the website, such as information about services, policies, FAQs, and documents. By training the model on this site-specific data, the chatbot becomes better equipped to provide accurate, context-aware responses to user queries. This customization ensures that the chatbot understands the language and terminology specific to the site and delivers relevant information based on the user's needs, improving the overall user experience.

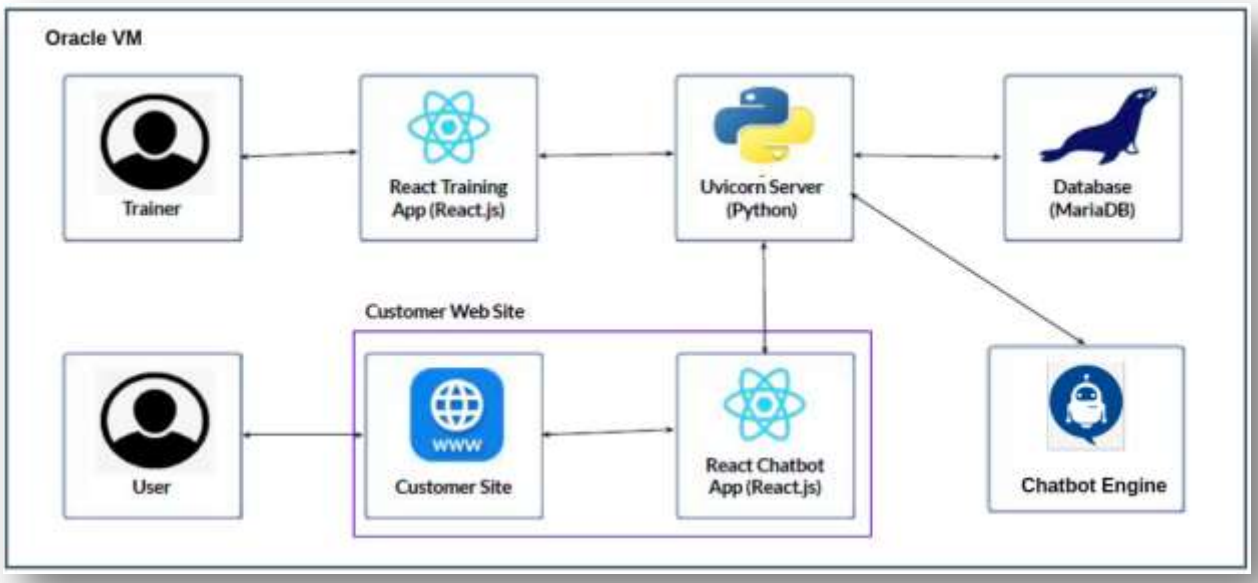
   The process starts with the user creating a client for the website they want to train the chatbot for. Then the user initiates the training process by selecting the "Full Site Training" option. This triggers the application to crawl the entire website, gather all the URLs, and scrape the content from the identified pages. The scraped data is then stored in a vector database to enable efficient querying and response generation. Additionally, the website provides options to train individual URLs and specific documents, allowing for more targeted and refined training. The user can also categorize URLs, helping organize the content for more focused search capabilities. This process ensures that the chatbot is well-equipped to understand and respond to user queries based on the site's unique structure and content.

2. **Chatting with the bot**

   In this phase, users engage with the trained chatbot through a conversational interface, where they can ask questions and retrieve information from the website. The chatbot provides accurate, context-specific responses based on the data it has been trained on, which includes details about the site's content, services, and documents. By leveraging the knowledge from the training phase, the chatbot offers precise answers to user queries, making it easier for users to access the information they need without having to navigate through the website manually. This phase enhances user experience by delivering quick, relevant responses to a wide range of questions.

AI Search Assistant for Complex Websites

Application Architecture:



Description:

The application provides a portal for training and deploying custom AI chatbots tailored for government websites. The frontend is built using React and offers a user-friendly dashboard where users can efficiently manage website training and chatbot interactions. The backend, developed with Python's FastAPI, seamlessly handles crawling, data extraction, and processing. This architecture enables comprehensive site training, document management, and chatbot integration to enhance user engagement and data retrieval.

To begin with, users need to create a client for a website by entering essential details; such as the client name, the display name for the bot, and the target website URL for scraping. Once this setup is complete, users can initiate the training process by selecting the “full-site training” option. The backend application uses Beautiful Soup to crawl the entire website, collecting all available links within the specified domain and scraping the content from each page. The scraped data is processed and stored in a structured file format, which is later indexed into Qdrant, a vector database optimized for fast and efficient content retrieval. Metadata such as URLs and training statuses are recorded in MariaDB to ensure proper tracking and management.

Users can access trained document pages by clicking on a client within the dashboard. This allows them to view all the documents processed during the training phase. Additionally, the application provides flexibility for training individual URLs or uploading specific documents directly. Users can

AI Search Assistant for Complex Websites

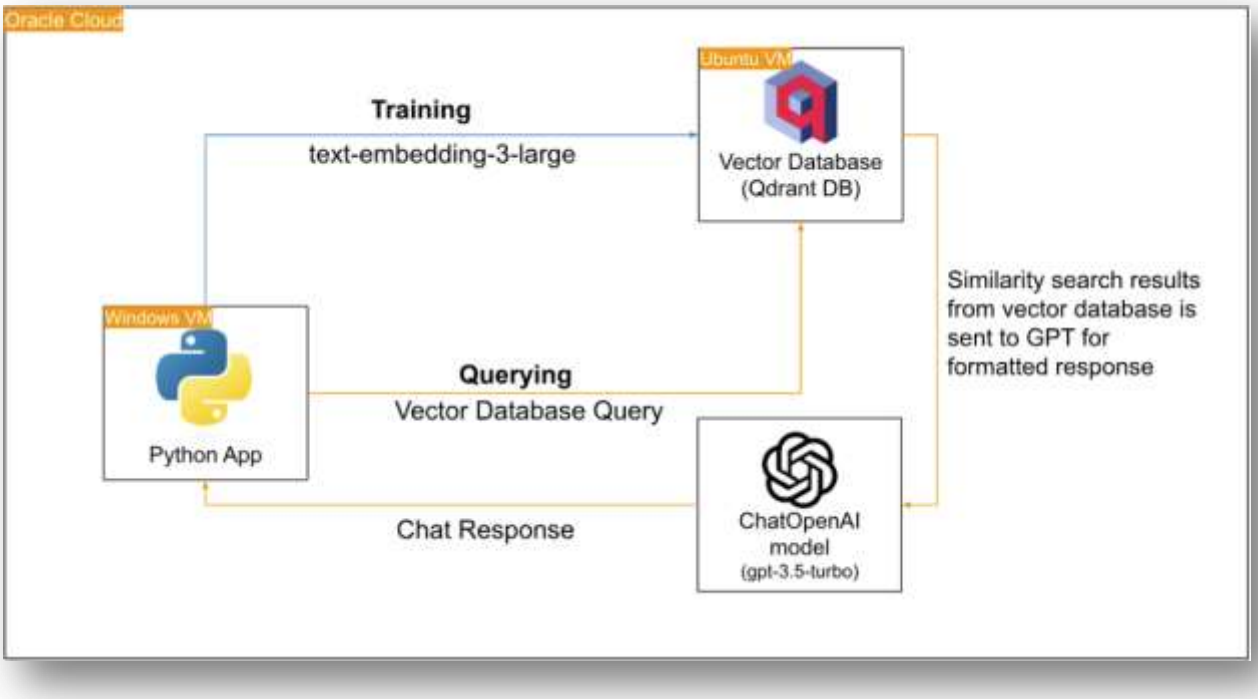
also create categories and assign them to documents, enabling better content organization and more focused data exploration.

The portal supports role-based access management, allowing admins to assign users to specific websites and provide them with roles tailored to their responsibilities.

Once the chatbot is trained, users can interact directly through the chat icon available on the dashboard against the corresponding client. For seamless integration into third-party applications, the portal offers a generate script feature that allows users to embed the chatbot into external web applications.

The application runs on Oracle windows VM.

Chatbot Engine Architecture:



The chatbot engine architecture is designed to deliver efficient and context-aware responses through advanced AI-driven processes. It seamlessly integrates vector search technology with powerful language models to handle complex user queries and provide accurate information retrieval.

AI Search Assistant for Complex Websites

During the training phase, the textual data extracted from websites and documents undergoes preprocessing and embedding. The system leverages **Text-Embedding-3 Large** to convert unstructured content into high-dimensional vectors, capturing the semantic relationships between words and phrases. These embeddings are then indexed and stored in **Qdrant**, a robust vector database optimized for similarity-based searches and high-performance data retrieval.

When a user interacts with the chatbot, the engine performs a **similarity search** within the relevant vector database collection. This search efficiently identifies data points that closely match the user's query, ensuring that the most relevant context is retrieved. The retrieved context is then passed to the **OpenAI language model**, which formats and tailors the response based on the query and underlying content context.

Finally, the generated response is presented to the user in a seamless conversational format, ensuring clarity and accuracy. This architecture enables the chatbot to maintain responsiveness, handle large datasets efficiently, and provide meaningful answers tailored to user needs

Variables impacting performance of the application:

- Huge size of data
- Long processing time
- Load on RAM for processing

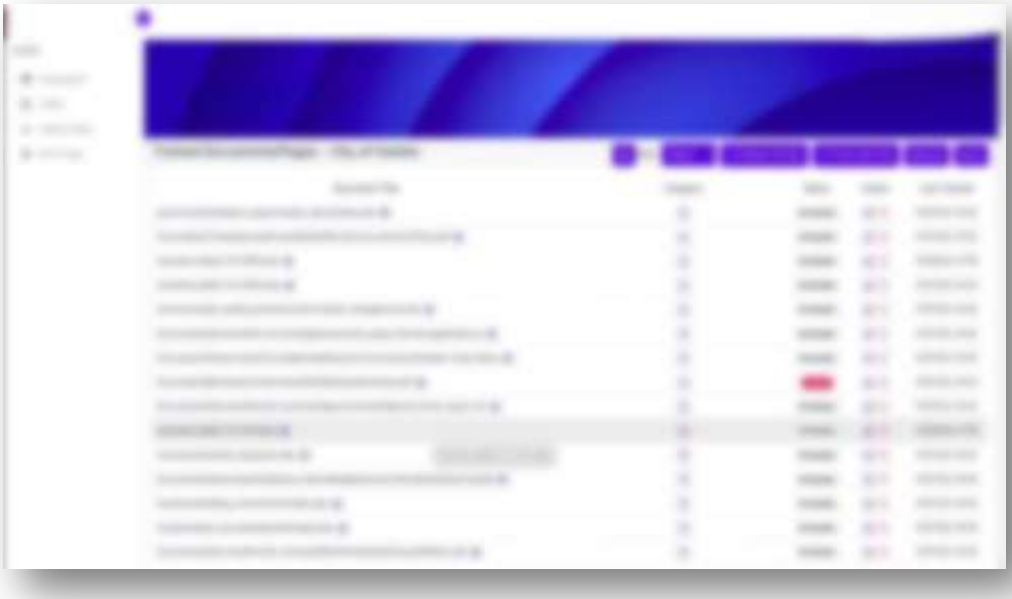
Key measures to overcome optimization challenges:

- Handling multiple website training in parallel
- Utilizing multithreading for simultaneous processing of documents to reduce processing time
- Using memory maps to store data on disk swap and load the data on demand

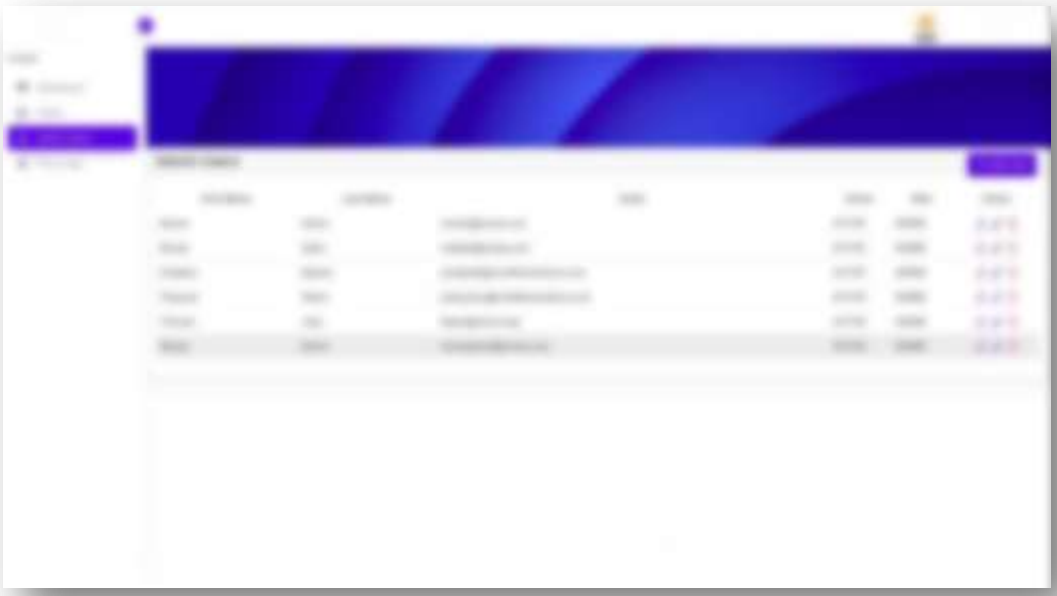
AI Search Assistant for Complex Websites

Screenshots:

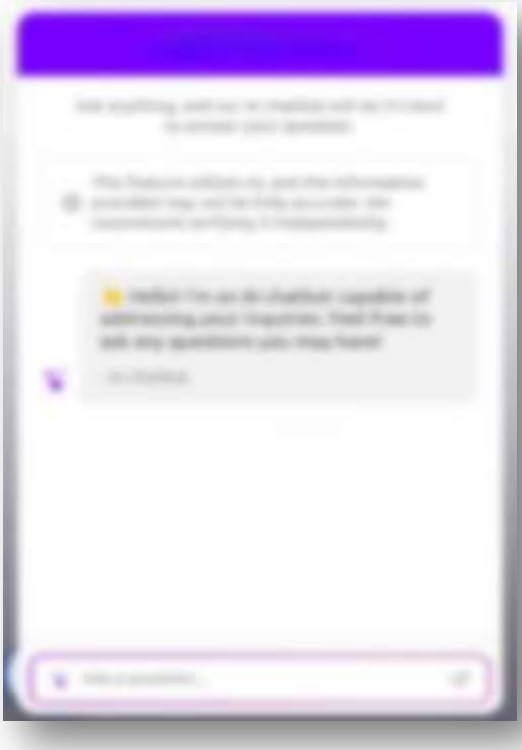
Training app



AI Search Assistant for Complex Websites



Chatbot



AI Search Assistant for Complex Websites

Analytics: