**Final Project Report – NestQuest**

**Krutal Patel, Sai Kumar Baddham, Likhith Kumar, Vishnu Priya Thanda**

MS in Data Science, UMBC

Data 690 – Natural language Processing

By Prof. Tony Diana

**Abstract**

The Nest Questemploys cutting-edge machine learning and natural language processing techniques to conduct home searches. The goal of this project is to create a user-friendly application that will allow people to find their ideal home. To deliver precise search results based on the user's location, price, availability, and other contributing criteria, the program makes use of the power of NLP and ML algorithms. A speech recognition system (Text-to-Speech) and a powerful database management system is used in the project to store and manage data for properties, their location, and their costs. The NestQuest, which has the potential to transform how people search for homes by making the process more convenient and efficient, is explained in this research article along with its approach and expected results.

**Introduction**

Automated chatbots are rapidly gaining popularity as customer support tools in various industries. These chatbots have voice-activated capabilities (Text-To-Speech), which enable users to listen to the output in audio format. The use of machine learning, a subset of artificial intelligence, plays a critical role in teaching these chatbots how to interpret and respond to users' queries accurately.

**Purpose**

The NestQuest main objective is to incorporate chatbots into the real estate sector. To generate leads, real estate brokers frequently invest a lot of time using customer information and quotations. Chatbot assistants can make this process easier by building a database of useful information that can be used to guide interactions with potential customers. Chatbots can improve engagement among buyers and agents. Also, this fact frees up agents to focus on more tasks, like negotiating deals and building relationships with clients, but they cannot take the place of the buyer-agent relationship.

**Dataset**

To evaluate the performance of the suggested approach, a dataset of fictitious data was produced. A random number generator was used to create the fake data, which had a specified distribution to mimic genuine data. The dataset had a variety of variables, including minimum credit score, square footage, amenities, and property type. The chatbot was able to test the algorithm's precision in seeing patterns and making predictions on fresh data by utilizing fictitious data. This strategy made testing possible without jeopardizing the confidentiality and privacy of real people. The program performed well on the fictitious dataset, demonstrating possibilities for use with real-world data in the future.

**Attributes**

The dataset utilized in the NestQuest comprises several variables that hold significance in the real estate sector. These variables include House ID, City, Locality, Price, Bedrooms, Bathrooms, Square Footage, Year Built/Age, Amenities, House Type, Sale Status, Minimum Credit Score, and Link. Each variable is selected to provide essential information about a property and enable the chatbot to provide accurate and relevant information to users.

The chatbot's ability to understand natural language nuances and integrate it with a powerful database management system will make the process of finding an ideal home more convenient and efficient for users. The chatbot will use the information provided by users to deliver precise search results based on location, price, availability, and other contributing factors.

**Methodology**

Our project utilizes natural language processing (NLP) and machine learning (ML) algorithms to analyze user commands and deliver precise search results based on the user's location, price, availability, and other contributing criteria. NLP is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between humans and computers using natural language. NLP algorithms are used to understand, interpret, and generate human language.

Here is a step-by-step breakdown of the method for setting up chatbot over our own data:

1. Data Ingestion: This step involves loading your data from arbitrary sources to text in a form that is used downstream. The sub-steps involved in this process include:

A picture containing diagram, text, font, plan

Description automatically generated

* Data Loading: We have loaded csv documents using CSV loader function of LangChain.
* Split chunks: To make the text more manageable for language models, it is chunked into smaller segments. It is necessary since language models have a limit to the amount of text they can handle.
* Text embeddings: Next, each chunk of text is embedded with a numerical representation. This is important since we only want to select the most relevant chunks of text for a given question, and this idea can be achieved by finding the most similar chunks in the embedding space.
* Vectorstore: Finally, the embeddings and documents are loaded into a vectorstore for an efficient retrieval of the most similar chunks. In our project, we utilized OpenAI's embeddings and a FAISS vectorstore.

1. Data Querying: Once we have ingested your data, we can query it using ChatGPT to generate responses to questions. The sub-steps involved in this process include:

A picture containing diagram, sketch, line, plan

Description automatically generated

LangChain uses a three-step process to generate responses for chatbot queries.

* First, it looks up relevant documents based on the embeddings and vector store created during data ingestion.
* Next, it generates a response using a language model that considers the stand-alone question and relevant documents.
* Finally, it condenses the question prompt by combining the chat history and new question into a single standalone question. This is important for allowing follow-up questions and improving the user experience.

The language model used to power the chatbot can be easily substituted to other models supported by LangChain or even a custom wrapper. In our project, we used the OpenAI LLM GPT 3.5 turbo model with our API key.

1. User Interface

Streamlit UI: The Streamlit library is used to create a user interface that allows users to interact with the chatbot. The UI is designed to be intuitive and user-friendly, with clear prompts for input and output.

1. TextToSpeech Functionality:

The textToSpeech functionality is integrated with the chatbot to provide an audio output of the generated responses.

Finally, the project can utilize a cloud server to host the application to provide scalability, dependability, and availability. The cloud server allows the chatbot system to handle multiple user requests at once without affecting the performance of the system.

**Results**

Overall, the methodology of the Voice-Activated Home Finder project uses NLP and ML algorithms to create a user-friendly application that can help people find their ideal home.

Here is a snapshot of how our product looks!

A screenshot of a chat

Description automatically generated with medium confidence

A screenshot of a chat

Description automatically generated with medium confidence

**Conclusion:**

In Conclusion, the project combines two crucial components, intent identification, and leading questions, to provide the most pertinent solutions. A dataset of fictitious data was produced to evaluate the performance of the suggested approach, and the chatbot performed well, demonstrating possibilities for usage with real-world data in the future. The chatbot's ability to understand natural language nuances and integrate it with a powerful database management system will make the process of finding an ideal home more convenient for users. Overall, the NestQuest project has the potential to transform how people search for homes by making the process more efficient and user-friendly.

**References**

1. Yvann. (2023, May 5). *Build a chatbot on your CSV data with Langchain and OpenAI*. Medium.

<https://betterprogramming.pub/build-a-chatbot-on-your-csv-data-with-langchain-and-openai-ed121f85f0cd>

1. LangChain. (2023, February 10). *Tutorial: CHATGPT over your data*. LangChain. <https://blog.langchain.dev/tutorial-chatgpt-over-your-data/>
2. Zillow, Inc. (n.d.). *Rental listing - 3449 rentals*. Zillow. <https://www.zillow.com/homes/for_rent/>
3. Home Assistant. (n.d.). *Dialogflow*. Home Assistant. <https://www.homeassistant.io/integrations/dialogflow/>