Creating a Domain-Specific Voice Chatbot as a Virtual Tour Assistant

Yi-Hsiu Lin
National Yang Ming Chiao Tung University, Department of Computer Science

ABSTRACT

In developing a chatbot as a virtual tour assistant for Pei Gui Hall, a tourist attraction in Chiayi County, I designed a Chinese language model capable of answering domain-specific questions. The chatbot also features voice input/output and an avatar interface to enhance user engagement.

The chatbot operates in two phases. First, a fixed-base retrieval mechanism calculates the similarity between user inputs and previously collected Q&A pairs, returning the most relevant response. If this approach does not provide an adequate answer, the RAG mechanism is employed, using pre-prepared domain knowledge to guide the language model in generating precise responses. This design ensures that users receive accurate and contextually relevant information during their visit.

1. Introduction

1.1 Motivation

As participants in the "Chiayi Cultech +1" creativity competition, we were tasked with proposing a project that integrates technology within the cultural spaces of Chiayi County. We chose Pei Gui Hall, a historical site in Hsin Kang Township, Chiayi County, as our theme. In executing the technological aspect of this project, I aimed to design a chatbot to serve as a virtual tour assistant for Pei Gui Hall. This Chatbot is a language model capable of answering domain-specific questions about Pei Gui Hall and Hsin Kang Township, enhancing visitors' experience with informative and interactive guidance.

1.2 Objective

With a primary aim of promoting local culture, the proposed chatbot is specifically designed for conversations in Chinese. Beyond answering domain-specific questions about Pei Gui Hall, this chatbot facilitates voice interactions and incorporates an avatar interface to enhance user engagement. In other words, it is equipped with the ability to receive user's voice messages, execute natural language processing, and then simulate an avatar speaking with a synchronized voice.

2. Related Works

There have been several domain-specific chatbots designed for FAQs in various domains.

In paper [1], a chatbot is designed to provide answers to any query based on a dataset of FAQs, utilizing Artificial Intelligence Markup Language (AIML) and Latent Semantic Analysis (LSA). Template-based and general inquiries, such as greetings and common questions, are handled using AIML, while more complex questions are addressed with LSA to ensure accurate responses. In [2], it finds the best matched answer from a predefined knowledge database. The user's query first undergoes pre-processing steps such as tokenization and stopwords removal. Then, it performs feature extraction based on n-grams and TF-IDF to extract keywords. Finally, it computes the cosine similarity between the query and the

entries in the knowledge database to retrieve the best matched answer. And in [3], it fine-tuned a cdQA-suite with additional closed-domain documents. However, this approach suffers from reduced accuracy and makes significant mistakes when handling longer texts.

Typically, chatbots are divided into text-based and voice-based categories. Text-based chatbots require users to type their input, whereas voice-based chatbots allow users to ask questions using their voice, making interactions more convenient and closely mimicking real human interactions. This project focuses on the latter. In [3], the application was developed with Voice to Text functionality using the Speech Recognition API. For text to voice conversion, the Speech Synthesis API was used. In [4], HTML5 Speech Recognition API is used to convert user speech into a JSON object. Information is then extracted from this JSON object as input for the chatbot. The chatbot's output is processed by HTML5's Speech Synthesis API, which breaks sentences into words, converts written text into phonetic form, determines intonation and rhythm, and combines phonetic details with tone. Finally, sound is generated by selecting optimal units from HTML5 Speech Synthesis API's acoustic database, returning a voice file.

An avatar is a visual depiction of an individual, appearing in either three-dimensional or two-dimensional form. It has been found to increase the degree of the smoothness of speaking to the partner [5]. In [4], a 3D avatar-based chatbot is developed using the Living Actor API to incorporate facial expression features. It achieves rendering an avatar whose gestures and lip movements synchronize with the audio reply, making it look like it is naturally speaking.

3. Methodology

3.1 Dataset

Here, the data I need are question-and-answer pairs related to Pei Gui Hall. However, since there are no FAQs on its official website, I have to create the dataset myself.

Firstly, I crawled all the necessary textual information from the website, such as 'Architecture of Pei Gui Hall', 'Directions and Parking', and other information that tourists might be interested in. This scraped data will be stored in a PDF file.

Next, I used ChatGPT to generate question-and-answer pairs based on the collected file. The prompts covered various aspects, such as history, architecture, transportation, etc., with at least 20 pairs for each aspect. Then, I filtered out the reasonable ones. The final dataset is saved as a CSV file.

3.2 Proposed Architecture

In this section, I explain the design details of the chatbot. To implement a voice-driven virtual tour assistant, the proposed architecture includes a speech recognition module for converting speech to text. Following this is the main component of the chatbot, responsible for natural language processing and

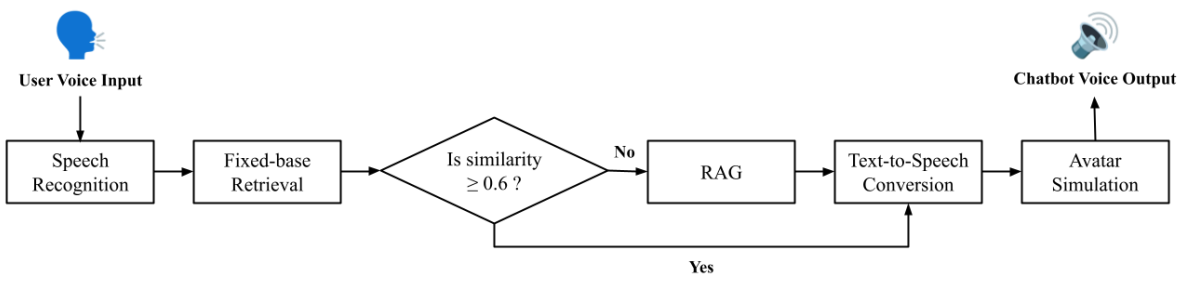

Fig. 1. Flowchart.

generating responses. Next, there is a text-to-speech conversion module. Finally, an avatar is simulated. The entire process is depicted in a flowchart, as shown in Fig. 1.

3.2.1 Speech Recognition

This step, also called Speech to Text, captures the user's speech through a microphone and converts it into text. I used Google's Speech Recognition API for the conversion, specifically targeting Mandarin Chinese for recognition.

3.2.2 Response Generation

The response generation is divided into two stages. First, a fixed-base retrieval will be performed. This involves calculating the similarity between the input query and the questions in the CSV knowledge base containing Q&A pairs. If the similarity exceeds a certain threshold, the corresponding answer will be returned. Otherwise, Retrieval-Augmented Generation (RAG) will be used, allowing the language model to generate an appropriate response based on the PDF knowledge base.

Fixed-base retrieval uses previously collected Q&A pairs. All texts undergo tokenization and stopwords removal, followed by feature extraction using TF-IDF. The input query is then compared with these questions in the vector space to compute similarity scores, returning the answer with the highest score. Only if the score exceeds 0.6 is the answer considered valid and returned; otherwise, the system moves to the next stage. Additionally, since the text is in Chinese and involves proper nouns related to the tourist site, I predefine stopwords and a dictionary to achieve better tokenization outcomes.

RAG is a technique for answering domain-specific questions by using pre-prepared domain knowledge to constrain the language model's responses. In this case, the knowledge base consists of information that was previously crawled and mapped to a vector space. The language model I use is developed by the TAIDE team, which aims to create a dialogue engine tailored to Taiwanese language and cultural characteristics. When a query is received, the system matches it with the most similar paragraphs in the knowledge base and sends these, along with the query, to the language model in order to provide accurate responses.

3.2.3 Text-to-Speech Conversion

In contrast to the Speech-to-Text process, after obtaining the answer, it needs to be converted back from text to speech. I tried two methods for this conversion: Google's Text-to-Speech API and a Python package. In the end, I chose the Python package because it does not require generating an MP3 file, and it allows for customization of speech rate, volume, and other properties.

3.2.4 Avatar Simulation

At this step, since I'm not familiar with producing dynamic animations, I used other tricks to simulate a character speaking. The steps mainly involved asking AI to draw two images: one depicting the avatar with an open mouth and the other with a closed mouth position. These images are then alternated during voice playback and synchronized appropriately with accompanying text.

4. Results

4.1 ChatBot Interface

The chatbot interface displays an avatar, and the user controls it using a keyboard. When the user presses any key, the microphone activates, allowing them to start speaking and asking questions. Once the system generates a response, the avatar simulates speaking, and the response is delivered through voice output. These

steps repeat until the user presses 'q' to exit.

Additionally, the dialogue box on the screen displays corresponding text, such as the user's query in the form 'You: <input> ?'. After processing the input, it shows 'Assistant: <output>'. This presentation resembles a conversation with an animated character and also enhances user understanding. Fig. 2 shows the two display states of the avatar.

4.2 Limitation

One limitation of this project is the lack of synonym handling. If the user's wording differs from the predefined knowledge base, even with the same meaning, the similarity score can be very low, often triggering the second stage of the RAG mechanism.

Another limitation is that the avatar currently simulates talking by alternating between two images with different mouth positions, which is less natural compared to actually rendering an avatar. Additionally, the synchronization between the avatar's movements and the voice is achieved using threading, which may result in slight timing discrepancies between the two.

5. Conclusion

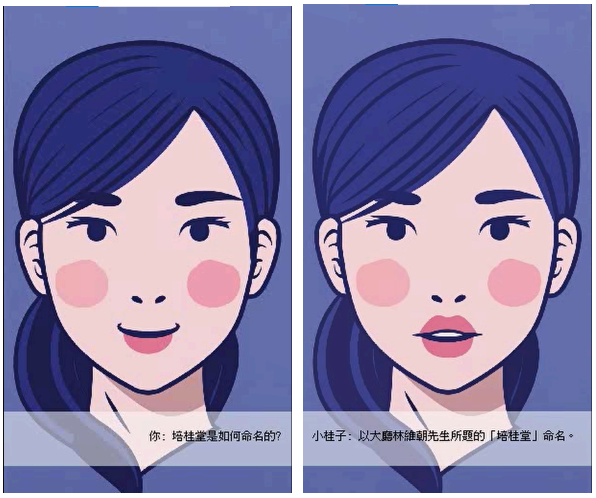

Fig. 2. Two states of the avatar. Left: thinking; Right: talking.

In this project, I implemented a Chinese chatbot that serves as a virtual tour assistant for Pei Gui Hall. It not only answers domain-specific questions but also enhances user interactions through voice and animated simulations.

Future work includes defining a synonym dictionary to improve similarity calculations. Additionally, exploring more advanced methods for rendering the avatar could lead to more natural simulations and interactions.

References

[1] Ranoliya, B. R., Raghuwanshi, N., & Singh, S. (2017, September). Chatbot for university related FAQs. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1525-1530). IEEE.

[2] Athota, L., Shukla, V. K., Pandey, N., & Rana, A. (2020, June). Chatbot for healthcare system using artificial intelligence. In *2020 8th International conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO)* (pp. 619-622). IEEE.

[3] Athikkal, S., & Jenq, J. (2022). Voice chatbot for Hospitality. arXiv preprint arXiv:2208.10926.

[4] Kurniawan, A. A., Fachri, W. E., Elevanita, A., & Agushinta, R. D. (2015, October). Design of chatbot with 3D avatar, voice interface, and facial expression. In 2015 international conference on science in information technology (ICSITech) (pp. 326-330). IEEE.

[5] Tanaka, K., Nakanishi, H., & Hiroshi, I. (2015). Appearance, motion, and embodiment: unpacking avatars by fine‐grained communication analysis. *Concurrency and Computation: Practice and Experience*, *27*(11), 2706-2724.