

WanderVoice: Innovative in-car Voice Assistant

Yizhan Zhang

Department of Electrical Engineering
Columbia University
New York, United States
yz4703@columbia.edu

Yuchen Zhu

Department of Statistics
Columbia University
New York, United States
yz4396@columbia.edu

Xinyang Wang

Data Science Institute
Columbia University
New York, United States
xw2964@columbia.edu

Abstract—This paper introduces WanderVoice, an innovative in-car AI voice assistant designed to enhance the driving experience by providing real-time travel information and suggestions. Unlike traditional voice assistants focusing solely on navigation, WanderVoice acts as a travel chatbot and a source of contextual travel services. The service is built on top of Amazon Web Services. By integrating search engines into the application, users will receive real-time recommendations on hotels, concerts, and many other events based on driving conditions. In combination with large language models, WanderVoice aims to redefine in-car assistance.

Index Terms—Amazon Web Services, Large Language Models, Retrieval-augmented Generation, API, Vue.js

I. INTRODUCTION

In the realm of automotive technology, traditional in-car voice assistants mainly focus on setting navigational goals and listening to users' instructions, which can effectively accomplish some hands-free tasks. However, these services have limited functionalities because they cannot provide tour guidance or potential real-time, contextually relevant suggestions for the user.

Illustrating this limitation, services like Google Maps offer comprehensive navigation assistance but fall short in delivering contextual destination insights. Inquiries about specific interests yield results more suited for stationary browsing rather than for an in-transit, auditory experience. When asking the “best museum in New York City,” it responds with a list of places in the form of text and images. This form is not convenient under driving conditions. Similarly, traditional GPS systems are restricted to route searching, lacking the capability to engage interactively with the user's travel-related curiosities.

Contrastingly, tour guide websites or apps can provide information about local attractions and historical sites. For example, washington.org develops a comprehensive online platform to provide information for various aspects of visiting Washington D.C. including events, attractions, deals, and accommodation. However, users have to spend more time to text and survey on computers to find out their interests. These resources are not tailored for instant access or advice during the journey, thereby not catering to the needs of drivers seeking spontaneous trip information.

Recognizing these gaps and the growing demand for enhanced and customized in-car assistance, our objective is to develop WanderVoice, an advanced AI-driven voice assistant

specifically designed for automotive use. WanderVoice is envisioned to serve dual purposes: firstly, as a travel chatbot, addressing queries related to the trip, and secondly, as a provider of real-time services and suggestions, tailored to the user's current journey. This innovative approach aims to redefine the in-car assistant experience, making travel more informative, interactive, and enjoyable.

II. USER INTERFACE

A. Login Service

If users have registered in the application, they can use their username and password to sign in. Otherwise, they need to create a new account. Interface of the login service is shown in Figure 1.

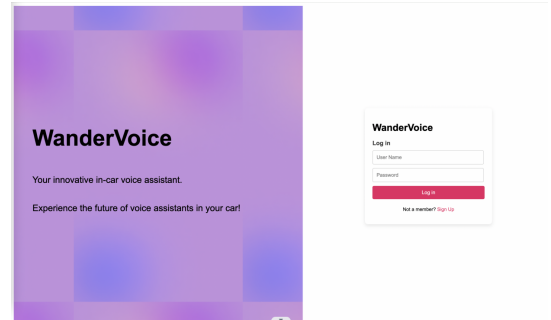


Fig. 1. Login Interface

B. Functionality Interface

Users can choose either Tour Guide or Recommendation service in this interface. Detailed description of the two services will be explained in the following two sections. Screenshot of the interface is in Figure 2.

C. Tour Guide Service

Tour Guide service is responsible for answering general questions, such as destination description, attractions in a city, and so on. The input format can be either text or voice, and the search result will be presented to users in both text and voice. Interface of the login service is shown in Figure 3.

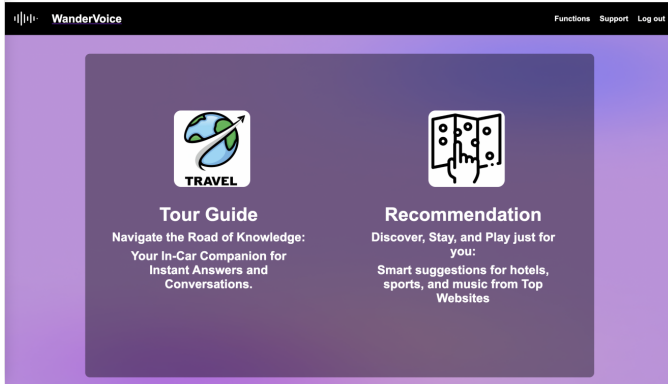


Fig. 2. Functionality

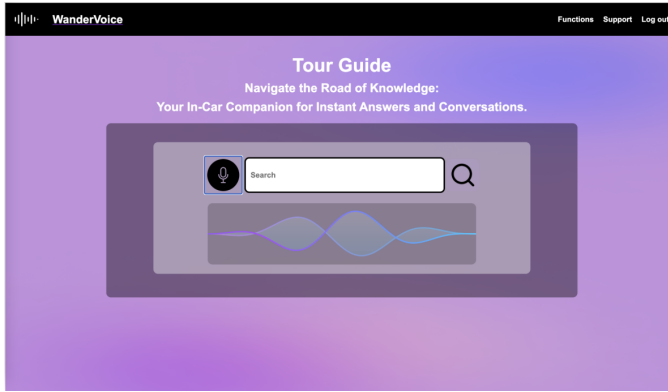


Fig. 3. Tour Guide Service

D. Recommendation Service

Recommendation service mainly focuses on questions about real-time route planning, hotel booking service, and event recommendation. Input and output format is similar to the Tour Guide service. The interface is shown in Figure 4.

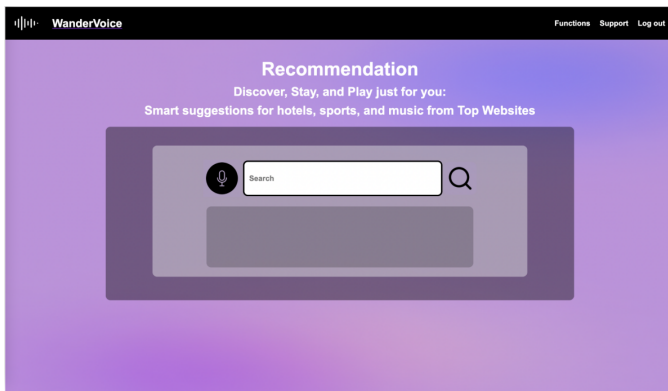


Fig. 4. Recommendation Service

III. SYSTEM DESIGN

A. Architecture

The architecture of WanderVoice is a sophisticated system designed to integrate various services and components to

deliver a comprehensive in-car voice assistant experience, as depicted in Figure 5.

The system is structured around several key components:

- **Login Service:** For registration, a Lambda function is used to save username and password in the database. For login verification, verify whether username and password match records in the database.
- **User Input:** If the input format is voice, upload it to AWS S3 bucket and use AWS Transcribe service to generate text based on the voice. Otherwise, directly pass the text input through API Gateway.
- **Tour Guide Service:** Leverage OpenAI's LLMs to respond to users' queries with pre-defined prompts.
- **Recommendation Service:** Integrate several business APIs into the chatbot to respond based on search results.
- **Response to Front-end:** Return the text response to the front end. Meanwhile, apply AWS Polly service to convert text into audio files, save it in S3 bucket, and send to the front end.

B. Detailed Design

In this part, we will elaborate on the detailed implementation of core functions of the application. To begin with, when users click on the microphone button in the front end, the voice will be recorded and uploaded to an S3 bucket B_1 . Whenever a new MP3 file is uploaded, a Lambda function LF_1 will be triggered to use AWS Transcribe to turn the audio into text.

After receiving text version of users' queries, a new Lambda function LF_2 will call APIs from Amadeus and Ticketmaster. Amadeus API allows developers to collect real-time housing information. Housing information is relevant to rating, location, and amenities of hotels. Ticketmaster API allows developers to collect real-time event information. Event information is about description, time, and ticket about events like music, sports, and so on. Based on those collected information, retrieval-augmented generation (RAG) [1] is implemented, where LangChain is used as the building block on top of the vanilla ChatGPT. In this process, background information is encoded into embedding and integrated into the text-generation function of ChatGPT. Then, the Lambda function LF_2 will call API from ChatGPT and respond to users' queries.

In the last step, we present the response in two formats to users. One format is text, which is directly passed to the front end through a Lambda function. The other one is voice, which is transformed from the original text using AWS Polly with target language and voice. In the front end, text will be displayed in the response part of the page, and audio will be played after users click on the play button.

IV. CODE DEVELOPMENT

A. Front-end Code

1) *Overview:* The front end development primarily focuses on the user interface for the functions introduced above, involving the design, framework, behaviours. We built front-end code and deployed on AWS S3 bucket and configured it as

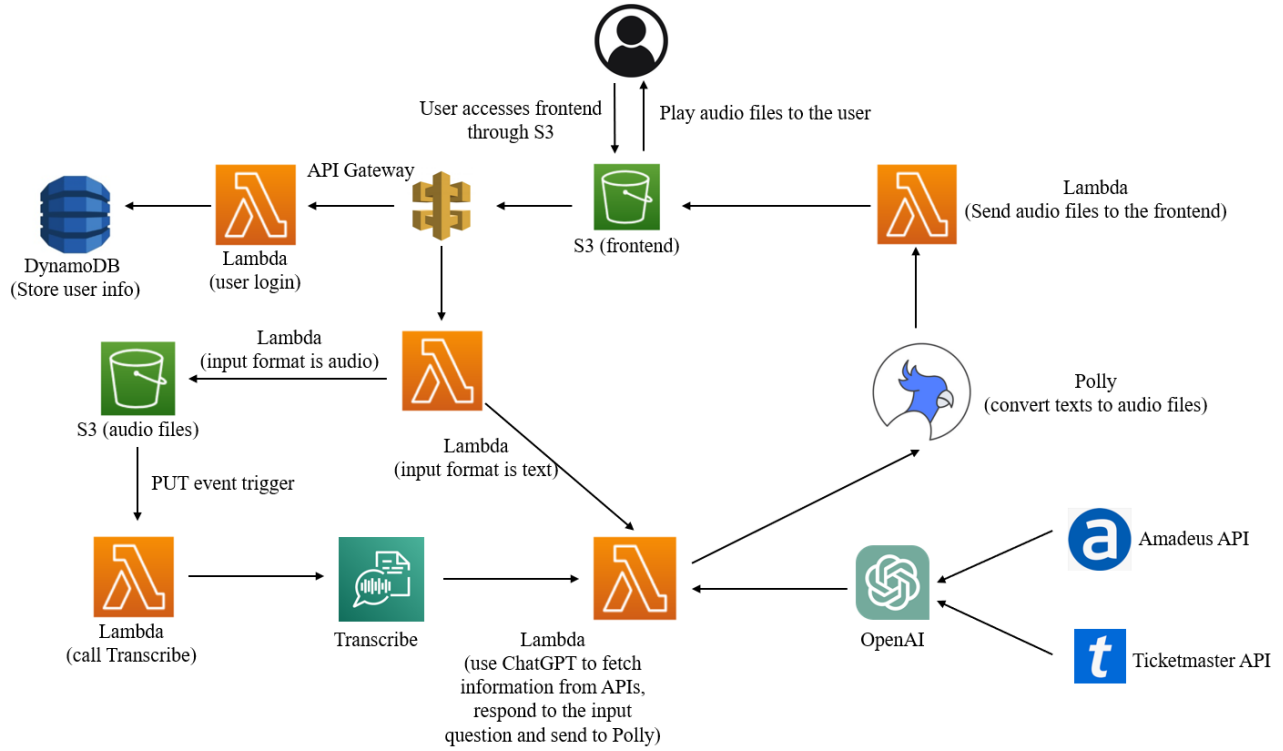


Fig. 5. Architecture Diagram of WanderVoice

a static website, so that we can enter the website from specific website name.

2) *Framework and Tools*: For our web pages, We selected Vue.js as the framework to engineer the front end code. Due to its simplicity, flexibility and ease of deployment on S3, we used it to build a component-based codebase. And it contains essential front-end technologies including HTML for webpage structure, CSS for styling, and JavaScript for interactivity. And we used 'recorder-core', one of the JavaScript libraries for audio recording in web applications.

3) *Code Structure*: Based on the functions, we built up components including login page, sign-up page, navigation bar, functions navigation page, and two function pages, and used router to jump between pages.

It starts with main.js and jumps to login page. In login page, we had input checks in front end, rather than sending to back-end and checking whether the inputs are valid. For example, it is checked whether both the username and password input boxes are filled as button is clicked, and also password should have a minimum length of 6. And then we used axios to send the message to AWS API Gateway, and the page will jump to functions navigation page if success response is returned. In the functions navigation page, this component mainly works as introduction and navigation of functions, and two image buttons works as router to two function page.

The Tour Guide and Recommendation components have similar structure. We set two button to allow two methods to input and query, which are audio and text. Once mic button is

clicked, the record on the page starts, and the wave is shown. Audio is stored as blob of mp3 and Recorder from recorder-core was used. Users should click the search button to upload both two methods of input by AWS API Gateway. Once the page gets the text response, it will immediately display the text. While for audio response, it will store and show a play button. Users can click the button to play the audio.

4) *Deployment*: Vue CLI provides a method to compile and build a Vue.js project into static files and generate a 'dist' file folder. We uploaded the folder to S3 bucket and configured the index document as index.html so users can access the website from the web name provided by AWS.

B. Chatbot Code

Core functions of WanderVoice was built based on AWS Lambda functions. They were responsible for different tasks, mainly focusing on login, input processing, data scraping, chatbot, and text-to-speech functions.

1) *Login Function*: There are two files about this functionality: *signup_service.py* and *login_service.py*. For *signup_service.py*, it is invoked when users try to register from the front end. If the username has never been used and the password satisfies pre-defined format, a successful status code will be returned and a new account will be created. Username and password of the new account will be saved in a DynamoDB database in AWS. Otherwise, an unsuccessful status code will be returned and the user needs to input a different username. For *login_service.py*, it is responsible

for verifying the username and the password. By searching the database in DynamoDB, the Lambda function will check whether the password is correct for the given username. If correct, users can enter the application, otherwise they need to log in again.

2) *Input-processing Function*: The *input_process.py* file is the input-processing function and triggered by PUT event of an MP3 file. After receiving the MP3 file, the function will connect to Transcribe client, initialize the transcription job, and define the target S3 bucket for storing the transcribed text.

3) *Chatbot Function*: First part of *chatbot.py* is the data scraping function. It is triggered by PUT event of an txt file, the content of which is the transcribed text. The function will call Amadeus and Ticketmaster APIs based on the current timestamp and retrieve real-time data about hotel booking information and upcoming events. Restricted by the usage of those APIs, the current version of application only supports searching within New York City. Second part of *chatbot.py* implements the chatbot function. Given the searched result mentioned above, this function will call OpenAI API with a prompt to get the response in text.

4) *Text-to-speech Function*: Text-to-speech function is implemented in *store_text.py*. After receiving the text format of response. It will connect to AWS Polly client to take the response and transform it into voice. We selected the common voice for in-car voice assistant and English as the target language. For the text output, it will be directly passed to the front end; for the audio output, it will be stored in another S3 bucket and fetched by the front end after completion.

V. RESULTS

Upon deployment, WanderVoice was subject to a series of real-world tests to evaluate its performance. In the following subsections, we presented queries and corresponding responses given by the application. Besides, we tested the running time for services.

A. Test on Tour Guide Service

Query 1: What are the top five must-visit landmarks in New York City?

Response 1: The top five most visited landmarks in New York City are Times Square, Central Park, the Statue of Liberty, the Empire State Building, and the Brooklyn Bridge.

Query 2: Could you provide a brief history of the Brooklyn Bridge as we drive across it?

Response 2: The Brooklyn Bridge is a historic bridge located in Coti. It has been an important transportation route for many years, connecting different parts of the town. The bridge is known for its architectural beauty and has become a popular landmark in the area.

B. Test on Recommendation Service

Query 1: Are there any special cultural events happening in New York City this weekend?

Response 1: Yes, there are several special cultural events happening in New York City. Here are a few:

1). New York Knicks vs. Milwaukee Bucks: This basketball game will take place on December 23, 2023, at Madison Square Garden. Tickets are on sale now.

2). Music For Enophiles: This music event will be held at The Bowery Electric on January 25, 2024. Tickets are available for purchase.

3). Apollo Music Café: Blakley: Enjoy a night of music at the Apollo Theater on February 2, 2024. Tickets can be purchased online.

4). Rad Museum w/ Miso: This concert will take place on January 24, 2024, at a location to be announced. Tickets are currently on sale.

5). Balloon Museum - Let's fly (December): Experience the Balloon Museum on December 19, 2023, with various activities and exhibits. Tickets are available for purchase.

These events offer a range of cultural experiences, from sports to music and art.

Query 2: I'm looking for a boutique hotel in Manhattan that's pet-friendly. Any suggestions?

Response 2: I found several pet-friendly hotels in Manhattan for you. The DREAM Meatpacking District, Four Seasons New York, and Hyatt Times Square New York are all highly rated 5-star hotels that allow pets. These hotels offer amenities such as a spa and valet parking. Another option is The Mercer, which also has a 5-star rating and offers valet parking. If you're looking for a hotel with parking and valet parking, the Four Seasons New York is a great choice. All of these hotels are located in the vibrant city of New York and provide a comfortable stay for both you and your furry friend.

From sample tests above, it can be seen that WanderVoice can generate high-quality response based on users' queries. If needed, it can also integrate real-time information into the response, which indicates its promising functionality.

C. Running Time Analysis

We performed ten tests on the Tour Guide service and the Recommendation service to keep track of the response time of text generation and audio generation processes. Mean and standard deviation of the response time is shown in Table I.

TABLE I
PERFORMANCE COMPARISON OF TIME SPENT ON DIFFERENT SERVICES.

Time (s)	Text output	Voice output
Tour Guide	4.6433 \pm 0.9367	35.3640 \pm 1.6500
Recommendation	11.8950 \pm 1.0921	42.1667 \pm 2.5166

Based on test statistics in the table, it can be found that the time for generating voice output is significantly longer than that of text. It is reasonable because voice-generation process requires additional resources like AWS Transcribe, Polly, and S3. Besides, running time of Recommendation service is longer than that of Tour Guide service, which is caused by additional usage of Ticketmaster and Amadeus APIs.

To improve users' experience when waiting for response, we added an interactive function to display the generation process and comfort users that the system is still operating. Screenshot for this function is in Figure 6.

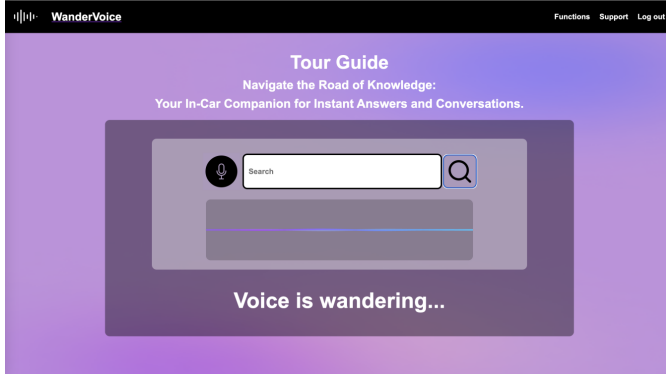


Fig. 6. Recommendation Service

VI. CONCLUSION

WanderVoice represents a step forward in the evolution of in-car voice assistants. Combining AWS services with OpenAI's LLMs, the platform's architecture, has proven effective in delivering a user-friendly and informative travel assistant. The results from the deployment show that WanderVoice successfully fulfills the dual roles it was designed for: acting as a knowledgeable travel chatbot and providing real-time, personalized suggestions and services.

In the current market, there is a marked demand for enhanced in-car voice assistants that offer more than just navigational aid. WanderVoice's ability to provide contextual information about destinations, coupled with its real-time recommendation services, sets a new standard for what drivers can expect from in-car technology.

Future iterations of WanderVoice could see improvements in the form of increased database comprehensiveness, refined natural language understanding for even more nuanced queries, and integration with additional APIs to broaden the scope of real-time services offered. As the technology progresses, the potential for WanderVoice to become an indispensable travel companion for drivers is immense.

REFERENCES

- [1] Anantha, R., Bethi, T., Vodianik, D., and Chappidi, S. (2023). Context Tuning for Retrieval Augmented Generation. arXiv preprint arXiv:2312.05708.