JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Smart Assistant for Automobiles with NLP based Personalised Speech-to-Speech chatbot

Mr. Nitin Singh Kushwah[1],

[1]AI & DS, Centre of AI, MITS Gwalior, M.P.

Abstract

With real-time GPS tracking, efficient route mapping, and an automated accident alert system that notifies emergency services, the Smart AI Assistant for Automobiles improves vehicle safety and navigation. For hands-free communication, it has a voice-to-voice AI assistant that enables it easy to access directions, vehicle status, and nearby service stations.

I. INTRODUCTION

Integrating advanced technologies is crucial for improving safety, navigation, and user interface in the constantly developing field of autonomous cars. With a range of clever features intended to guarantee a smooth driving experience, the Smart AI Assistant for Automobiles seeks to transform this market. Effective journey tracking is made possible by this technology, which uses real-time GPS-based location fetching to precisely calculate the position of the car and display it on an interactive map.

The system's accident alert feature, which instantly sends an email to the closest emergency services in the event of an incident, is one of its primary features. Furthermore, the system has a voice-to-voice AI assistant that uses sophisticated natural language processing to offer hands-free navigation support, car status updates, and recommendations for nearby services.

A. Key Features

1. Location fetching using GPS
2. Showing it on map
3. Accident Alert mail to neaest department
4. Personel AI assisstant to guide (Voice-to-Voice control)
5. Continous Monitoring
6. Setting path to desired location

B. Necessary Libraries

```
import asyncio
import winsdk.windows.devices.geolocation as wdg
import folium
import json
import os
from IPython.display import display, HTML
import smtplib
from email.mime.text import MIMEText
import google.generativeai as genai
import speech_recognition as sr
import pyttsx3
from flask import Flask, render_template, request
```

1. Asynchronous programming for non-blocking operations is handled by asyncio; it's particularly helpful for AI replies and GPS data retrieval.
2. wdg, or winsdk.windows.devices.geolocation [1] obtains the current position of the vehicle by retrieving real-time GPS-based location data.
3. Folium: Creates and shows interactive maps to visualize routes and show service stations or gas stations in the area.
4. Route and geolocation data handling need the ability to parse and manipulate JSON data from local files or APIs.
5. OS: Ensures smooth system-level activities by managing environment variables and file paths.
6. For debugging or demonstration reasons, IPython.display shows HTML information or visual features (such as maps) directly in a Jupyter Notebook.

7. By connecting to the SMTP [8] server, smtplib sends emails containing accident alerts.
8. The email notification delivered to emergency services is formatted using email.mime.text.
9. Google's Generative AI is used to integrate chatbot capabilities and improve voice-to-voice communication (google.generativeai) [2].
10. In order to communicate with the AI assistant, speech_recognition (sr) recognizes and translates voice commands into text.
11. The AI assistant can respond with voice commands thanks to pyttsx3, which translates text to speech.
12. Flask: Constructs the backend web server that serves HTML templates, handles HTTP requests, and powers the project's frontend.
13. render_template & request: This makes it easier to handle form or user input and render web pages.

II. LITERATURE REVIEW

With an emphasis on improving safety, navigation, and driver assistance, artificial intelligence (AI) integration in cars has grown significantly in popularity in recent years. A completely integrated AI assistant with real-time accident detection, route optimization, and tailored support is still lacking, despite the fact that current systems have made progress in tackling individual areas.

The authors of the paper "Vehicle Tracking Using GPS and GPRS" illustrated the value of precise location tracking in transportation systems by presenting an affordable system for real-time vehicle monitoring utilizing GPS and GPRS technology.[3] Effective vehicle fleet management and tracking were made possible by the system's successful transmission of positioning data to a central server.

The authors of the study "AI-based Personal Assistant Using Machine Learning and Natural Language Processing" examine the development and application of an AI assistant that uses ML and NLP techniques to enable interactions that are similar to those of a human [4]. The technology provides insights into contextual awareness and chatbot personalization by concentrating on comprehending and reacting to user inquiries via speech and text. Although this method offers a strong basis for intelligent personal assistants, it does not particularly address emergency management or automotive integration, which are features that the proposed Smart AI Assistant for Automobiles seeks to include for real-time vehicle support.

Sudhakar Reddy M., Vyshnavi, Raju Kumar, and Saumya (2020) describe a virtual personal assistant (VPA) that uses Python programming and artificial intelligence to manage calendars, to-do lists, and emails using voice commands[5]. The system interprets user inputs and responds with synthetic speech by combining speech recognition and natural language processing (NLP). Notably, the authors stress improving speech-to-text accuracy by utilizing machine learning approaches to combine lip movement detection with voice recognition. By enhancing the assistant's comprehension of users with varying accents, this method seeks to make it more useful in everyday situations.

Fitro et al. (2018) tackle the computational difficulties of finding the shortest path in Geographic Information Systems (GIS) by combining a node combination technique with Dijkstra's algorithm to minimize memory usage and guarantee the derivation of the most efficient route between two points [6]. Using data from Taman Subdistrict, Sidoarjo, East Java, Indonesia, which consists of 17 nodes and 72 vertices, the study computes distances based on latitude and longitude values obtained via the Google Maps API. This combined method not only minimizes memory consumption during the route search process, but also ensures the optimality of the solution, which makes it especially pertinent for use in real-time vehicle navigation systems.

Gaikwad et al. (2021) thoroughly examine a range of accident detection and alarm systems in their study, highlighting the vital necessity of prompt notification to emergency services in order to lower the number of fatalities [7]. The study looks at a number of approaches, such as using accelerometers, GPS, GSM, and Arduino-based systems, to identify car crashes and quickly alert emergency contacts and the appropriate authorities. The authors point out the benefits and drawbacks of each strategy, pointing out that while technology like accelerometers are good at detecting abrupt impacts, they can also result in false positives. In order to improve the dependability and effectiveness of accident detection systems, the study emphasizes the significance of integrating

various sensors and communication technologies. This is especially pertinent when creating intelligent in-vehicle safety helpers.

A. Gaps in Existing Technology

Although each of these systems handles navigation, safety, and personal help separately, there isn't an AI assistant that does all three at once.

1. Real-time notification of personal contacts and emergency services for accidents.
2. Voice-to-voice personal help for questions about cars.
3. dynamic route optimization for the detection of adjacent service stations and gas stations.

By providing a multipurpose, AI-powered solution that improves driver convenience and safety, the suggested Smart AI Assistant for Automobiles seeks to close this gap.

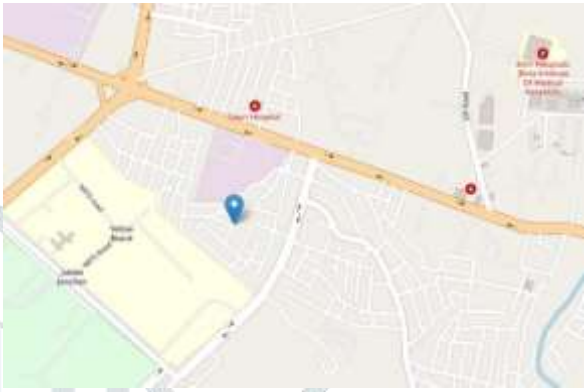
III. METHODOLOGY

A. Speech-to-Speech chatbot

Module for Speech Recognition: To record and identify user voice input, the system makes use of speech recognition. The function adjust for ambient-noise calibrates the microphone to take background noise into consideration. The Google Speech-to-Text API is used to extract the identified text. Generating AI Responses: An NLP model that has already been configured receives user input. The request tells the AI to react in 100 words and function as a car helper [2]. The assistant replies with a backup message if the question has nothing to do with cars. Module for Text-to-Speech (TTS): The AI-generated response is transformed into voice via pyttsx3. A female voice (ZIRA) with pace and volume adjustments is set up in the TTS engine. Handling Errors: handles situations in which the API request is unsuccessful or the speech is ambiguous.

B. Location Fetching and route optimization

Devices uses winsdk.windows.Geolocation via get_geoposition_async() to asynchronously acquire the vehicle's current GPS coordinates [1]. The function controls the event loop for asynchronous execution and deals with permission failures. Data Management for Service Centers: The locations and names of neighboring service stations and gas pumps are saved in a JSON file. The json package in Python is used to load and process the data. Making Maps and Adding Markers: Folium.The map starts with a zoom level of 15 and is centered on the location of the car. The position of the car is indicated by a "YOU" marker. Folium iterates over the service center coordinates from the JSON file and uses green icons to mark them on the map.Marker(). We may also obtain the path to the closest service center by utilizing the ORS API.



C. Emergency Mail Feature

Use geopy to reverse geolocation and determine a location. GPS coordinates of the car (location_cordinate) are transformed into a human-readable address by Nominatim. The closest address is retrieved from the coordinates using geolocator.reverse(). Creating Email Content: The following is included in the accident alert text: Detected location address. The accident site's latitude and longitude. Email Configuration and Authentication: an initial connection to the Gmail SMTP server (smtp.gmail.com on port 587) is established using smtplib.SMTP.[8] Transport Layer Security (TLS) is used by starttls() to guarantee a secure connection. Transmission of the Email: MIMEText prepares the email including the accident information. When an accident is identified, emergency contacts automatically receive an email with the accident facts, including the location and GPS coordinates.

IV. CONCLUSION

The Smart AI Assistant for Cars effectively combines a number of technologies to improve driver convenience and vehicle safety. A clever voice-to-voice chatbot to help with vehicle-related inquiries, real-time location retrieval and map representation, accident detection with automated email alerts to emergency contacts and authorities, and effective identification of nearby service stations and gas stations with suggested routes are all features of the system.

A. Future Work:

1. Edge AI for Improved Accident Detection By using edge computing, accident detection may be handled locally, cutting down on response times and reliance on cloud servers.
2. Connectivity to IoT Devices Proactive maintenance is made possible by connecting the system to IoT sensors in cars to track fuel levels, tire pressure, and engine condition.
3. Using Local Data Storage in Offline Mode establishing offline features to process and store data locally in the event that an internet connection is not accessible.
4. Multilingual Chatbot Support adding multilingual support to the chatbot's capabilities in order to serve a wide variety of drivers worldwide.
5. Usage of Service mailing system to handle large bundkes of mail instead of SMTP.

V. REFERENCES

[1] Microsoft, "Windows.Devices.Geolocation Namespace," Microsoft Learn, [Online].

[2] Google Cloud Skills Boost, " Introduction to Generative AI Learning Path", [Online]

[3] K. Satyanarayana, M. P. Kumar, and G. S. Reddy, "Vehicle tracking using GPS and GPRS," *Proceedings of the ISPRS Commission VI Mid-Term Symposium*, vol. XXXVI, part 6, pp. 208–212, 2006.

[4] A. Mahapatra and R. Sarangi, "AI-based Personal Assistant Using Machine Learning and Natural Language Processing," in *Proceedings of the First International Conference on Computational Intelligence and Knowledge Economy (ICCIKE 2021)*, CEUR-WS, vol. 2889, pp. 1–6, 2021.

[5] Sudhakar Reddy M., Vyshnavi, Raju Kumar, & Saumya. (2020). *Virtual Assistant Using Artificial Intelligence and Python*. Journal of Emerging Technologies and Innovative Research (JETIR), 7(3), 1116–1119.

[6] Fitro, A., Bachri, O. S., Purnomo, A. I. S., & Frendianata, I. (2018). *Shortest Path Finding in Geographical Information Systems Using Node Combination and Dijkstra Algorithm*. International Journal of Mechanical Engineering and Technology (IJMET), 9(2), 755–760.

[7] Gaikwad, S., Khanna, M., Kothari, S., & Kudale, A. (2021). *Systematic Literature Survey on Accident Alert & Detection System*. CEUR Workshop Proceedings, 2889, 1–6.

[8] Al-Radhi, S. (2019) **"***Automatic Accident Detection and Notification System*", International Journal of Computer Science and Mobile Computing (IJCSMC).

[9] Hoy, B. M. (2018) **"***Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants",* Medical Reference Services Quarterly.

[10] Khanna, S. (2015) **"***Implementation of SMTP Protocol in Email Transmission"***,** International Journal of Computer Science and Information Security (IJCSIS).

[11] Kumar, D. (2018). *Review on SMTP Protocol for E-Mail Systems*. International Journal of Research and Analytical Reviews (IJRAR), 5(1), 127–132.

[12] Jurafsky, D., & Martin, J. H. (2021) **"***Speech and Language Processing (3rd Edition)".*