Demo: Towards a Conversational LLM-based Voice Assistant for Transportation Applications

Sasan Jafarnejad, Abigail Berthe–Pardo, Raphaël Frank
Interdisciplinary Centre for Security, Reliability and Trust (SnT)
University of Luxembourg
29 Avenue J.F Kennedy, L-1855 Luxembourg
{sasan.jafarnejad, raphael.frank}@uni.lu, abigail.berthe@hotmail.fr

***Abstract*—Conversational assistants based on large language models (LLMs) have spread widely across many domains, and the automotive industry is keen to follow suit. However, current LLMs lack sufficient understanding of geospatial data; in addition, timely information, such as weather and traffic conditions, is inaccessible to LLMs. In this demo, we present an in-car assistant capable of verbally communicating with the driver, and by utilizing external APIs, it can answer questions related to routing, finding points of interest, and is aware of the local weather and traffic conditions. The assistant, including a customizable speech synthesizer, is accessible through a graphical user interface that facilitates experimentation by simulating the change in time, origin, destination, and location of the car.**

***Index Terms*—LLM, AI, In-Car Assistant, V2C**

I. Introduction

As large language models (LLMs) such as ChatGPT gain mainstream adoption, the automotive industry is racing to integrate these powerful systems into vehicles. For example, BMW has announced that its *Alexa LLM* powered assistant can answer car-related questions [1]. TomTom is teasing an assistant powered by OpenAI's GPT and TomTom's APIs [2]. Cerence Inc. introduced an automotive-specific large language model dubbed CaLLM, however, its capabilities were not published [3]. Despite all these announcements, traditional players are notoriously slow in bringing new technologies to the market. These examples show the appetite in the industry for an LLM tailored for automotive and mobility applications.

Although the industry has shown interest, existing models do not possess a thorough understanding of geospatial data. In addition, LLMs require complementary technologies to utilize real-time information such as weather and traffic conditions.

In this work, we introduce KITT[1] (Knowledge-based Intelligence for Transportation Technologies). KITT is a voice assistant designed for in vehicle communication, it uses an LLM to enable vocal interactions between the car and its passengers. KITT can drastically reduce communication needs by operating locally, except when reaching out to third-party data sources. KITT integrates Speech-To-Text (STT) to listen and transcribe user queries. It uses LLM to process and generate response to user queries. Then uses Text-To-Speech (TTS) functionalities to playback the response. KITT uses *function-calling* (described further in Section II) to integrate several external APIs to obtain real-time information, making it a comprehensive tool to create a conversational assistant for cars.

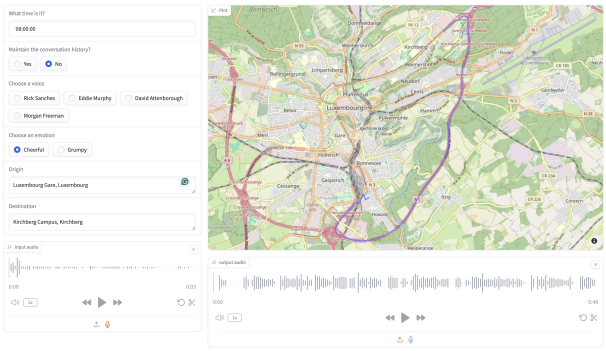

Fig. 1. A snapshot of the KITT's demonstration graphical user interface.

In this demo, we offer to interact with KITT using a graphical user interface (GUI) application, as shown in Figure 1. The application allows the user to simulate various scenarios by letting them set the origin and destination of the ride, the location of the car, the time of day, and the choice of voice.

II. System Architecture

The architecture of KITT is shown in Figure 2. We start with the graphical user interface (GUI), which is built with Gradio[2] but it is primarily designed to facilitate the demo, however, we recognize that many conversational queries could benefit from visual feedback in addition to the voice response. For a voice assistant like KITT the user interacts with voice. Voice is a suitable interface for the automotive environment, as it is less distracting than most alternatives. Therefore, STT and TTS are extremely important. To convert speech to text, we use Whisper, a model introduced in 2022 by OpenAI [4], while more models have been introduced since then, Whisper has the widest support, and we benefit from availability of heavily optimized implementations of Whisper. Whisper is trained on a massive dataset of 680,000 hours of multilingual and multitask data, which leads to improved robustness to accents, background noise, and technical language. When a response is ready, it must be converted to speech. There are

[1]KITT's source code is available at https://github.com/sasan-j/kitt

[2]Gradio http://gradio.app

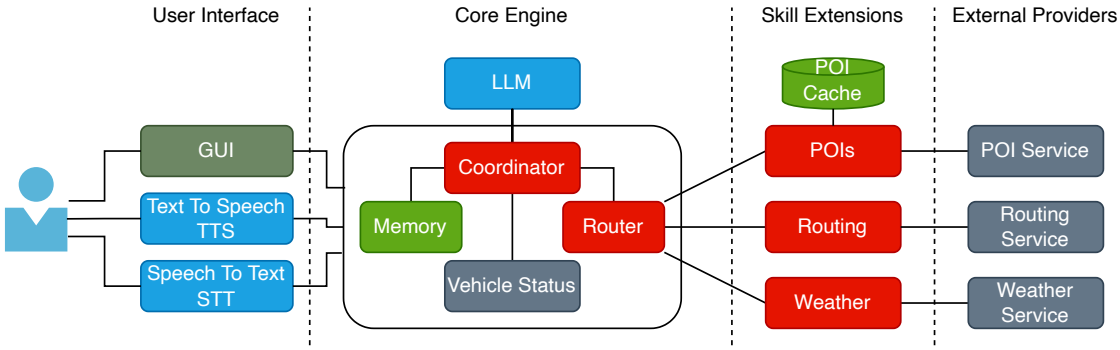

Fig. 2. Architecture of KITT. We use color to semantically highlight the functionality of each block. Red blocks primarily contain application code, green blocks are data storage, gray represents data sources, light blue blocks contain ML models, and dark green is the GUI application.

several options available for TTS, in KITT we use XTTS-v2[3] from Coqui, which is a multilingual model, and has zero shot capabilities on voice cloning, allowing the user to easily adapt the assistant to any desired voice, by only providing as little as 6 seconds of voice sample. In addition, XTTS-v2 can also clone emotions. XTTS is an improved version of YourTTS [5].

The most important part of KITT is what we call *Core Engine*, which could be described as the brain of KITT. To build KITT on top of an LLM, we use primitives from LangChain[4], which, by providing abstractions, facilitates working with LLMs. We control the history of conversation by limiting it to last n queries, so that it does not overflow the context window of the LLM. When the history grows beyond the limit, we use the LLM to summarize the previous conversations. Vehicle status provides information, such as current location, to the LLM by placing it in the prompt. To enable the model to interact with external APIs, we use function-calling. First, we provide a description of all available functions (APIs calls) to the LLM by placing them into the context. The LLM parses the input to determine if a function needs to be called, it extracts relevant parameters and generates a structured output specifying the function name and arguments. Therefore, for a given query, e.g. *Is there a Pizzeria nearby?*, the LLM may decide that it must call a function to obtain nearby restaurants. We call the function with specified arguments, obtain the results (restaurants) and place them into the prompt then execute another query, and let the LLM find the correct answer, i.e. the closest or highest rated Pizzeria. This technique enables KITT to access any arbitrary external data source. The LLM needs to be fine-tuned for function-calling, that is why we picked NexusRaven-V2 [6] for use in KITT. NexusRaven is instruction-tuned on Meta's CodeLlama-13B-instruct [7]. We use TomTom[5] for POI, traffic and routing related queries, and use the Weather API[6] for weather.

KITT runs on a consumer GPU such as the NVIDIA RTX 3090 24GB. The GUI application is reachable locally or over the Internet, therefore, suitable to run locally or in the cloud.

[3]Coqui ⓍTTS-v2 https://huggingface.co/coqui/XTTS-v2
[4]LangChain https://www.langchain.com
[5]TomTom https://www.tomtom.com
[6]Weather API https://www.weatherapi.com

III. Demonstration

The demo consists of a GUI application that allows the user to talk to the KITT voice assistant. The application allows the user to change the KITT's voice and emotion. The user can choose the origin and destination. We use an external provider to find a route from the origin to the destination and plot it on the map, and using a slider, we can simulate the movement of the vehicle along the route. This way, one can test scenarios, such as asking for POIs around the current location or along the route. We also let the user select the time of day to show that the predictions for the estimated time of arrival are adjusted according to the time of day.

IV. Conclusion and Future Work

In this work we presented KITT a LLM-based voice assistant, which has the ability to interact with external APIs. In future work we would like to expand KITT's capabilities by adding more skills. We also intend to explore various approaches to effectively embed geospatial data into the LLM to reduce the need to reach out to the Cloud.

References

[1] "Generative AI, Augmented Reality and Teleoperated Parking – (CES) 2024.." https://www.press.bmwgroup.com/global/article/detail/T0438824EN/?language=en.
[2] "Meet TomTom's in-car AI assistant." https://www.tomtom.com/newsroom/product-focus/meet-tomtom-in-car-ai-assistant/.
[3] "Cerence Pioneers Automotive-Specific LLM in Collaboration with NVIDIA, Powering the Future of In-Car Experiences | Cerence." https://www.cerence.com/news-releases/news-release-details/cerence-pioneers-automotive-specific-llm-collaboration-nvidia/, Dec. 2023.
[4] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022.
[5] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone," in *International Conference on Machine Learning*, pp. 2709–2720, PMLR, 2022.
[6] N. team, "Nexusraven-v2: Surpassing gpt-4 for zero-shot function calling," 2023.
[7] B. Rozière, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, R. Sauvestre, T. Remez, J. Rapin, A. Kozhevnikov, I. Evtimov, J. Bitton, M. Bhatt, C. C. Ferrer, A. Grattafiori, W. Xiong, A. Défossez, J. Copet, F. Azhar, H. Touvron, L. Martin, N. Usunier, T. Scialom, and G. Synnaeve, "Code llama: Open foundation models for code," 2024.