Computer Science & Computer Engineering

J.B. Hunt Mobile Virtual Assistant

Presented by: Brandon Cox, Caleb Fritz, Alicia Gillum, Anjan Poudel, Matthew Sij
University of Arkansas, College of Engineering: Computer Science

Introduction

The objective of this project is to save time for the drivers and add user friendliness to the Carrier 360 App. Because of easiness in controlling a device with a natural language rather than interacting with the screen, virtual assistance has become a crucial need of clients today to cope with quickly growing technology.

Adding virtual assistance features in Carrier 360 Mobile mobile application will allow drivers to search for and book loads without even touching their device. It will enable drivers to safely command the app to find a load while they are behind the wheel.

Purpose

The current process for booking new loads through the J.B. Hunt Carrier 360 Mobile Application requires that the driver manually enter key information throughout the booking and bidding process.

Because of the inherent danger of interacting with mobile devices and applications, drivers must be stationary to search, select, or bid on a new job.

This is a huge pain point for current and future users of the J.B. Hunt Carrier 360 Mobile Application, which aims to modernize freight services.

Methods

Client Design: The user interface must be intuitive, simple, and minimal in physical interaction. Consideration of the visual differences between various device types must be accounted for as well as the battery life of said devices.

Server Design: The server should be easy for new users to understand and should not impede users to the point to where they would prefer to use a touchscreen interface over a voice interface. Users should be able to search for loads by load number or search for loads using other parameters.

Conclusion

The goal of this project is to provide a virtual assistant that can help to automate tasks and improve safety for Carrier 360 mobile app users.

Integration testing has confirmed that the client-server architecture can process user utterances and return useful responses, though due to endpoint restrictions, must do so offline with fixture data.

Improvements to the system involve using test and production endpoints, analytics, implementing dialog backtracking, and enhancing TTS.

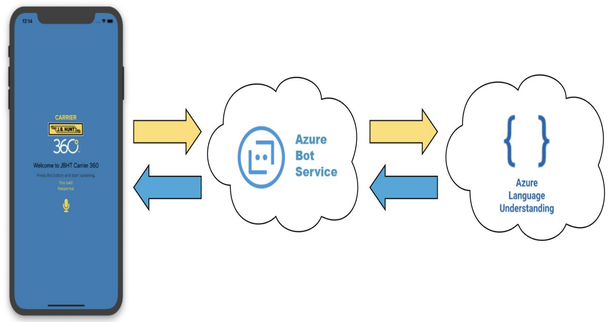

Figure 1. High-level representation of the interactions between the client mobile app, bot server, and LUIS NLP framework

Results

Client:

- Minimal, intuitive interface
  - Provides user utterances and bot server responses through on-screen text for the user to verify their utterance and the response from it in addition to...
  - Text-to-speech (TTS) capability to convey bot responses
  - Animated speech button to provide visual cues when the app is listening for utterances
- Automated dialog management
  - For multi-turn conversations, bot responses that require immediate user input automatically trigger the client to cue for an utterance
  - Once a user is done speaking, the client will automatically stop listening without another button press
  - Minimizes contact with the button to enhance safety

Server:

- A help response is provided to introduce the user to bot capabilities
- Supports load number search workflow
  - Users can search for a specific load number to...
    - View more information on that load
    - Book the load
    - Cancel and return to the main dialogue
- Supports route search workflow
  - Users may find loads based on parameters like origin, destination, equipment type, and maximum allowable empty miles to...
    - Choose from a list of possible loads returned from the search
    - View more information on that load
    - Book the load
    - Cancel and return to the main dialogue