2019

THE JOURNAL ON TECHNOLOGY AND PERSONS WITH DISABILITIES

Implementing the ATLAS Self-Driving Vehicle Voice User Interface

Julian Brinkley

Clemson University, School of Computing, Human-Centered Computing Division

Shaundra B. Daily

Duke University, Department of Computer Science

Juan E. Gilbert

University of Florida, Computer and Information Science and Engineering

julianbrinkley@clemson.edu, shani.b@duke.edu, juan@ufl.edu

Abstract

This article describes the implementation of the ATLAS self-driving vehicle voice user interface prototype. ATLAS has been designed to support the experiential needs of visually impaired self-driving vehicle users.

Keywords

Self-Driving Vehicles; Voice User Interface; Accessibility; Visual Impairment; Transportation

Introduction

The *Accessible Technology Leveraged for Autonomous vehicles System,* or ATLAS is a prototype human-machine interface intended to support the spatial orientation and navigation needs of self-driving vehicle users with visual disabilities. Within this report we describe the development of the system's voice user interface and share lessons learned during the development process. We argue that a review of our experiences in implementing the ATLAS VUI may prove beneficial for researchers who may face similar challenges in rapidly prototyping voice user interfaces.

Prototype Implementation

System Overview

The development of ATLAS was motivated by a desire to address the self-driving vehicle accessibility challenges suggested by the visually impaired survey respondents of our prior survey (Brinkley, Daily, et al.), focus groups (Brinkley, Posadas, et al.) and participatory design sessions. This report describes the implementation of the system's voice user interface specifically.

System Implementation

Though ATLAS is a single system, it has been designed around service-oriented design principles and utilizes a service-oriented, multi-tiered architecture. The core of ATLAS is a Microsoft Universal Windows Platform (UWP) application ("Intro to the Universal Windows Platform"), written in C#. Diverging from the approach of Asp.Net, Windows Presentation Foundation and WinForms, which required specific deployment environments, the UWP provides a core application-programming interface across devices. With this approach, a single application package can be installed on a range of

devices and can be made available to consumers through the Microsoft Store. Though it is necessary to tailor the application for different form factors and types of input, a single application may be deployed to a Microsoft Surface tablet, desktop computer, mobile device, Microsoft HoloLens, or other hardware provided that the Windows 10 operating system is present. The use of a UWP application was chosen over other approaches given the research team's familiarity with Microsoft .Net development, the decision to use C# for application logic, the desire to deploy the application to a Microsoft Surface tablet and the inherent compatibility of UWP applications with Microsoft's Language Understanding Intelligence Service ("LUIS: Language Understanding Intelligent Service") and Microsoft's Face API ("Face API Service Overview"); APIs used within the system for language understanding and affective analysis respectively.

Voice User Interface

Throughout the formative research activities participants indicated a significant desire for a robust voice user interface (VUI). Voice interaction in ATLAS was enabled through the development of a speech recognition and speech synthesis library written to augment the Universal Window's Platform existing speech capabilities, the development of a project-specific dialogue manager and the incorporation of language understanding as a service using a cloud-based language model.

Dialogue Strategy

Based on what was learned during a series of participatory design sessions, the high fidelity prototype was implemented to utilize a mixed-initiative dialog strategy (Cohen et al.). In a directed dialog strategy, the system initiates and directs all interaction. Using this approach, a system might ask a user a highly specific question (e.g. "What

time do you want to schedule the appointment?") and will anticipate a highly specific response (e.g. "1:30 pm"). In a mixed-initiative dialog strategy, at times the user has the initiative, and other times, the system will assume the initiative as needed to request missing information. Within ATLAS, first time users must complete a one-minute tutorial during which, the system maintains the initiative. At the conclusion of the tutorial, initiative is relinquished to the user who may initiate a system action using the "hot word" to activate the voice interface ("Atlas"). Depending upon the user's utterance the system will then assume the initiative as necessary. A user interested in learning the present time while the system is in standby for instance might say the hot word, "Atlas", to activate the voice interface followed by his or her request for the time. Given that the system uses mixed-initiative, the user could also simply say, "Atlas, what time is it?".

A sample dialogue is presented that demonstrates a mixed initiative interaction between the system *S* and a user *U* that is possible within ATLAS:

U: "ATLAS, I've been in an accident!" (The user has the initiative)

S: "You have been in an accident?" (The system assumes initiative)

U: "Yes!"

S: "Ok, you have been in an accident. Please stay calm and remain in your vehicle if it is safe to do so. I am contacting emergency services now. For your safety all exterior cameras and microphones have been activated. Is anyone injured?"

U: "Yes!"

S: "I have notified emergency services regarding potential injuries, they are on their way." (Initiative is returned to the user)

Given that the system uses a mixed initiative approach, participants of the prior design sessions suggested the presence of an audible cue to indicate when the system was expecting a user response. A similar cue is present when input is anticipated using the Job Access With Speech (JAWS, Freedom Scientific) and NonVisual Desktop Access (NVAccess) screen reading software; two popular accessibility tools widely used by persons with visual disabilities. A brief beep was incorporated whenever the system assumed the initiative and expected a response as demonstrated in this sample dialogue:

U: "ATLAS, drive" (The user has the initiative)

S: "Ok, where do you want to go?" (The system assumes initiative)

S: *beep*

U: "I want to go to Starbucks in Gainesville, Florida."

S: "Ok, you want to go to Starbucks in Gainesville, Florida?"

S: *beep*

U: "Yes"

S: "Ok, finding Starbucks in Gainesville, Florida. Please wait, this might take a moment."

Pervasive Dialogue Elements

The design of the ATLAS dialogue strategy also included the development of an error recovery strategy and a strategy for universal commands. ATLAS utilizes an approach to voice user interface error handling that combines escalating detail and rapid re-prompting depending upon the situation. In all cases, when the speech recognizer fails to return a recognition result within the specific confidence threshold (moderate confidence or higher) or the user fails to respond within a three second window, the

system re-prompts the user while repeating the question (e.g. “I’m sorry, what did you say?”). In certain situations, such as within the tutorial when users are gaining familiarity with the system, the user is given additional detail and specific directions on how to respond to the prompt (e.g. “I’m sorry I didn’t understand you. Try saying yes or no.”). Within the tutorial, when users are unable to respond to the practice prompt appropriately after three attempts (“Say, ATLAS what time is it?”), the system switches to a new prompt entirely (“It seems like you’re having trouble. Let’s try something else. Say, ATLAS what day of the week is it?”). At any point in a dialogue the system is able to respond to the universal commands of “Help”, “Exit”, “Quit”, “Emergency”, “Stop” and “Accident.”

Dialogue Management

A dialogue manager within a voice user interface determines what actions the system takes in response to user input. The dialogue manager within ATLAS is written in C# and takes as input either the raw user utterance from the speech recognizer or the user’s intent from the language understanding model. The dialogue manager performs one or more actions based either on keywords detected within the utterance (e.g. “Quit”, “Help”) or on the user’s interpreted intent based on processing the utterance through the cloud-based language model.

Language Understanding

Language understanding technology uses machine learning and a language model to enable an application to understand a user’s intent using the user’s own words. A system capable of language understanding might use a language model to determine that a user who says, “Find me a car for January 9th” wants to reserve a rental car on the 9th

day of the first month. ATLAS uses Microsoft's LUIS ("LUIS: Language Understanding Intelligent Service") language understanding service to host a language model in the cloud that maps user utterances, consumed as text input, to a user's intent (e.g. what action they want to perform). The ATLAS language model, Enteraxion Atlas v 1.1, contains 40 custom and prebuilt intents, processes utterances received as HTTP requests through the model and returns a response as a JSON object. The LUIS model also returns any relevant objects or entities that were detected during processing as well as a confidence score between 0.00 (no confidence) and 1.00 (highest confidence). A request to, "Find Starbucks in Gainesville, Florida" for instance returns the intent, "Places.FindPlace" with a confidence score of 0.86 and encyclopedia (Starbucks) and geography (Florida) entities. The resulting JSON object can then be consumed by ATLAS, which then takes action based on the content of the response.

Conclusion

Because voice interfaces do not require the motor skills needed for text input through a keyboard, their use may reduce the interaction barriers faced by persons with disabilities and older adults. The design of accessible voice user interfaces may still prove challenging however. Within this report we describe the implementation of the voice user interface of the ATLAS system, a prototype human-machine interface intended to support the spatial orientation and navigation needs of self-driving vehicle users with visual disabilities. We argue that our experiences in implementing the ATLAS VUI may prove beneficial for researchers who may face similar challenges in rapidly developing other accessible VUIs in both the self-driving vehicle and other contexts.

Works Cited

Brinkley, J., S. B. Daily, et al. "A Survey of Visually Impaired Consumers About Self-Driving Vehicles." *Journal on Technology And Persons with Disabilities*, vol. 6, Mar. 2018, pp. 274–83.

Brinkley, J., B. Posadas, et al. "Opinions and Preferences of Blind and Low Vision Consumers Regarding Self-Driving Vehicles: Results of Focus Group Discussions." *Proceedings of the 19th International SIG ACCESS Conference on Computers and Accessibility*, 2017, pp. 290–99.

Cohen, Michael H., et al. *Voice User Interface Design*. Pearson, 2004.

Freedom Scientific. *JAWS Screen Reader*. http://www.freedomscientific.com/Products/Blindness/JAWS. Accessed 12 Mar. 2017.

Microsoft. "Face API Service Overview." *Microsoft Azure*, 17 Mar. 2017, https://docs.microsoft.com/en-us/azure/cognitive-services/face/overview.

---. "Intro to the Universal Windows Platform." *Windows Dev Center*, 27 Oct. 2017, https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide.

---. "LUIS: Language Understanding Intelligent Service." *Language Understanding (LUIS)*, 2017, https://www.luis.ai/home.

NVAccess. "NV Access." *NV Access*, 2017, https://www.nvaccess.org/.