Adasa: A Conversational In-Vehicle Digital Assistant for Advanced Driver Assistance Features

Shih-Chieh Lin[1], **Chang-Hong Hsu**[1], **Walter Talamonti**[2], **Yunqi Zhang**[1], **Steve Oney**[1]
Jason Mars[1], **Lingjia Tang**[1]

[1]University of Michigan, Ann Arbor, MI
[2]Ford Motor Company, Dearborn, MI
{shihclin, hsuch, yunqi, soney, profmars, lingjia}@umich.edu, wtalamo1@ford.com

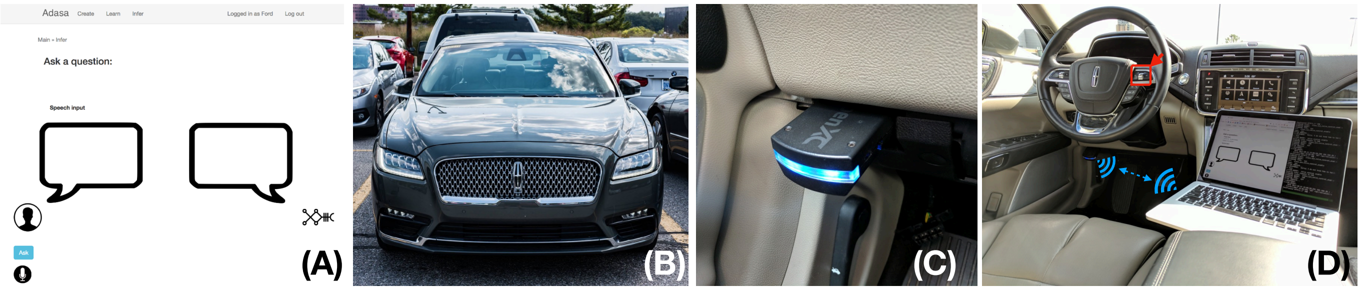

Figure 1: (A) The proposed digital assistant, Adasa, identifies user's questions or commands regarding ADAS features in human language and responds with answers or actions accordingly. (B) Adasa is integrated into a commercially available vehicle for evaluation and a real-world driving user study. (C) Adasa is able to access vehicle CAN signals via Bluetooth to conduct system diagnosis or system control. (D) Adasa setup, driver enables Adasa by pressing the button on the wheel to start the conversation.

ABSTRACT

Advanced Driver Assistance Systems (ADAS) come equipped on most modern vehicles and are intended to assist the driver and enhance the driving experience through features such as lane keeping system and adaptive cruise control. However, recent studies show that few people utilize these features for several reasons. First, ADAS features were not common until recently. Second, most users are unfamiliar with these features and do not know what to expect. Finally, the interface for operating these features is not intuitive. To help drivers understand ADAS features, we present a conversational in-vehicle digital assistant that responds to drivers' questions and commands in natural language. With the system prototyped herein, drivers can ask questions or command using unconstrained natural language in the vehicle, and the assistant trained by using advanced machine learning techniques, coupled with access to vehicle signals, responds in real-time based on conversational context. Results of our system prototyped on a production vehicle are presented, demonstrating its effectiveness in improving driver understanding and usability of ADAS.

UIST '18, October 14–17, 2018, Berlin, Germany

DOI: https://doi.org/10.1145/3242587.3242593

Author Keywords

Advanced Driver Assistance System; Automobile Interface; Voice Interface; Machine Learning

CCS Concepts

•**Human-centered computing** → **User interface design; Natural language interfaces;**

INTRODUCTION

Advanced Driver Assistance Systems (ADAS) have recently started to become widely deployed across the newer vehicle fleet. ADAS features are designed to either warn or assist in the control of a vehicle to help reduce the effects of human error while driving. For example, Lane Keeping System (LKS), which vibrates the steering wheel to alert drivers when their vehicles drift out of the lane, and Adaptive Cruise Control (ACC), which adjusts a vehicle's speed to maintain a certain distance from other vehicles, are two widely-used ADAS features.

While the available data on the influence of ADAS features is limited, some studies have estimated a promising influence on the real-world driving experience. One study estimates that ADAS may help drivers prevent 28% of all crashes if all new car purchases included ADAS [18]. The market for ADAS has grown significantly and nearly all major automakers (including Ford, Chrysler, BMW, and Audi) integrate ADAS into their vehicles. The market for ADAS is projected to continue growing: McKinsey & Co. researchers have predicted the ADAS market will double its size in the next three years, reaching $35 billion in annual revenue [13].

However, an estimated 73% of drivers with ADAS-enabled vehicles have not even attempted to use these features [5]. This is due to a number of factors [12, 23], including the fact that ADAS are relatively new and constantly evolving. Specifically, there are gulfs of evaluation [30] (i.e., drivers have difficulty assessing the state of ADAS, such as if they are activated) and gulfs of execution [30] (i.e., drivers are unsure how to activate and use ADAS). Because ADAS features could help the driver maintain control of their vehicle, it is important to build user interfaces that help to bridge these gulfs.

Building an effective interface for ADAS is challenging for several reasons. First, it is unclear what information drivers need to effectively use ADAS features. Second, the interface can not require any complex visual-manual interactions because asking the driver to perform such operations while driving could reduce safety. Furthermore, these are complicated systems whose behavior depends on a complex combination of system states and vehicle contexts. For example, ACC might appear to be deactivated because the feature is malfunctioning, the system is in a state where it is not used, or the feature is inactive and the driver must be able to distinguish between these three states. Finally, as we found in our analysis of customer inquiries, drivers often feel unclear what their roles entail when ADAS features are activated.

Currently drivers interact with ADAS features through controls on their steering wheel and indicators displayed on the dashboard. In this paper, we propose to add a third modality: a speech-based conversational interface for ADAS features. We designed and built **Adasa**, the first speech-based conversational interface for ADAS. Adasa's features are based on our analysis of over 9,000 conversations between drivers and Ford's customer service division and includes additional training data generated by crowd workers. We built Adasa upon the state-of-the-art conversational machine learning platform, Lucida [20], which allows drivers to interact with Adasa in unconstrained natural language in real-time. Drivers can simply ask questions or issue commands after enabling Adasa by pressing a single button on the steering wheel.

Adasa can handle *queries* that do not require the current vehicle's status (e.g., the meaning of a symbol on the dashboard), *system diagnostic* questions related to the vehicle and ADAS state (e.g., "why is my wheel vibrating?"), or *commands* to control ADAS features via natural language. We integrated Adasa into a commercially available vehicle and conducted a user study on 15 drivers in a real-world driving environment. In the evaluation, we perform quantitative and qualitative analyses of our user study based on prior work in evaluating information system [10, 11], which focuses on system quality, information quality and user satisfaction. The results of our study show that Adasa correctly identified, understood, and responded to over 77% or participants' unconstrained natural language commands about ADAS, which included questions about ADAS state and commands to precisely control their features. In addition, our user feedback demonstrates the effectiveness in improving the user understanding and overall driving experience using ADAS features (an average score of 8.9/10). Specifically, this paper makes the following main contributions:

- An analysis of over 9,000 discussions between drivers and customer service representatives from a major auto manufacturer about ADAS to better understand drivers' information needs.
- Adasa, an in-vehicle digital assistant, that allows the driver to ask questions or command and control in human natural language while driving to help drivers understand the features and improve the usability.
- An evaluation of Adasa deployed on a production vehicle, which demonstrates the effectiveness of the proposed system in improving the usability of these ADAS features.
- Insights gained through our user study in terms of how the digital assistant system design affects the overall user experience, and how traffic conditions impact user interaction.

RELATED WORK

In general, the number and complexity of systems available to drivers has increased significantly [22]. HCI researchers have investigated how to improve in-vehicle user interfaces for systems like driving assistance, infotainment, entertainment and car-integrated mobile devices [3, 14, 28, 36, 40].

User Interfaces for Cars

To help drivers use these systems without being too distracted while driving, different types of in-vehicle interfaces have been built and studied [15, 22, 25, 27, 31, 34, 42]. However, most of these interfaces have been visual or tactile. Kern et al. explore the design space of driver-based automotive user interfaces, including a set of inputs (e.g., button, touchscreen and pedals) and outputs (e.g., multi-functional display, digital and analog speedometer) modalities [22]. Visual user interfaces in vehicles have been investigated, but the results show that driving experience and behavior would be affected notably, and these interfaces may distract drivers from the primary driving tasks (i.e., driving and focusing on the traffic) [21, 35]. On the other hand, Ohn-Bar et al. investigate how gesture interfaces for the in-vehicle systems should be designed to improve the driving experience. They present the feasibility to have gestural interfaces deployed in the cars for a wide range of in-vehicle functionalities [31]. Lee et al. study the implications for drivers when using voice interfaces and touch interfaces on semi-automated systems and find that drivers who use the voice interface to control automated driving have lower nervousness and make fewer driving mistakes (e.g., road edge excursion) than those who use the touch interface [27].

In-vehicle Voice Interfaces

There is a large body of work showing that voice interfaces allow drivers to effectively focus on driving and the environment. Among all the interface modalities, voice interfaces help improve driving safety compared to other interface modalities [2, 17, 24, 26, 27, 28, 29, 38]. Researchers also find that voice interfaces affect human behavior, as well as emotion. Graham et al. evaluate users' experience of using voice interfaces to perform secondary tasks while driving. Despite the fact that voice interface is slower, less accurate, and leading to lower task performance, the results show that users still consider it easy to learn and logical, expressing the preference over manual interface [17]. In addition, prior works present that drivers spend more time keeping their eyes on the road when using a speech interface than a manual interface [2, 29]. From the industrial point of view, in-vehicle voice interfaces are now widely deployed in commercialized vehicles such as Ford Sync [6], Toyota Entune [9] and GM MyLink [7]. Even mobile devices are designed to be able to connect to in-vehicle infotainment and entertainment systems via CarPlay [1] or Android Auto [16]. These technologies leverage advanced speech recognition techniques to allow users to interact with systems like navigation or multimedia entertainment via voice commands. However, recent commercial products mostly rely on constrained speech (i.e., using specific terms or formats) which can be distracting while driving as it may require a higher cognitive demand than unconstrained natural speech. In fact, interacting with conversational systems using natural human language is still challenging and remains a crucial problem [19, 40, 41]. In this work, we employ advanced machine learning techniques to design Adasa and enable drivers to interact with vehicles in unconstrained natural language in real-time. None of these aforementioned speech-based systems are particularly designed for ADAS features, which we found to be a gap and may directly affect the drivers' in-vehicle experience while driving as more ADAS features are introduced.

UNDERSTANDING DRIVERS' INFORMATION NEEDS

To understand what information drivers need and what the critical problems are that prevents them from utilizing ADAS features, we first conducted an in-depth analysis on customer verbatims describing the real problems that the owners of vehicles equipped with ADAS features encounter. These verbatims collected by Ford Customer Service Division (FCSD) via customer call center services include over 9,000 cases provided for analysis consisting of owners of Ford Fusion, Ford Explorer, Lincoln MKS and Lincoln MKT equipped with ADAS features from 2013 to 2017. From these verbatims, we identify three primary question categories:

- **Division of driving responsibility between the driver and ADAS** – More than 4,400 out of the 9,000 (i.e., 48.9%) verbatims ask about the expected division of driving responsibility between the driver and the ADAS feature. For example, drivers complain LKS does not keep the vehicle in its lane after it has been switched on, which is because LKS only engages when the vehicle speed is higher than its operational threshold (i.e., 40 miles per hour) and the lane marking lines are visible.

- **Interface to activate ADAS features** – Over 2,000 cases (i.e., 23.1%) ask how to turn on certain ADAS features (e.g., LKS). This is primarily because drivers are not familiar with these features and they often have a hard time recognizing the buttons to turn them on and off. This demonstrates that there is a gulf of execution that prevents most drivers from utilizing ADAS features.

- **Meaning of instrument cluster iconography** – 1,300 out of 9,000 (i.e., 14.5%) verbatims ask about the symbols on the dashboard because quite a few ADAS features present different symbols dynamically depending on the real-time surrounding environment (e.g., a typical ACC shows a vehicle symbol on the dashboard only when there is a vehicle ahead) which also shows that most drivers encounter issues of understanding.

One could argue that auto manufacturers provide comprehensive information about these questions in the vehicle owner's manual. However, many drivers are still found to be confused about ADAS features for two reasons, thereby not being able to utilize them in their vehicles. First, it is tedious to read the printed user manuals due to their lengthiness (i.e., often hundreds of pages). Second, these questions should be addressed in real time, as opposed to conventional approaches where drivers could not and should not read user manuals while driving. Feedback from the Ford user experience design team further confirms these observations. Expecting all users to carefully read the owner's manual may be overly optimistic, and these 9,000 verbatims we analyzed have demonstrated that to be the case. Therefore, we conclude an improved and more natural user interface is required to improve the usability of ADAS features.

SYSTEM DESIGN

To enhance the current user interface for ADAS features and improve its usability, we present **Adasa**, an in-vehicle digital assistant based on the insights we gained from our analysis of over 9,000 real user verbatims. Adasa allows the driver to ask any question or command and control in natural human language while driving, and the assistant analyzes the human speech in real-time and combines it with the contextual information about vehicle status (e.g., what mode ACC is currently operating in) to provide responses accordingly in human speech.

In this section, we first formalize the design objectives for Adasa. We then present an overview of the components in Adasa, and walk through the life of an example query to illustrate the workflow of the system. Lastly, we present the details of the key hardware apparatus and software components.

Design Objectives

To improve the system through real-world testing and subjective evaluation beyond the insights we gained from analyzing the call center verbatims, we formalize the following items as the key design objectives of an in-vehicle digital assistant:

1. **Intelligent understanding** – The digital assistant is able to understand incomplete questions and identify out-of-scope queries properly.

2. **Real-time processing** – The digital assistant has the ability to process queries and respond with the requested information to drivers in real-time in a dynamic driving environment.

3. **Accurate responses** – The digital assistant can provide useful information regarding driver's questions and control the vehicle correctly.

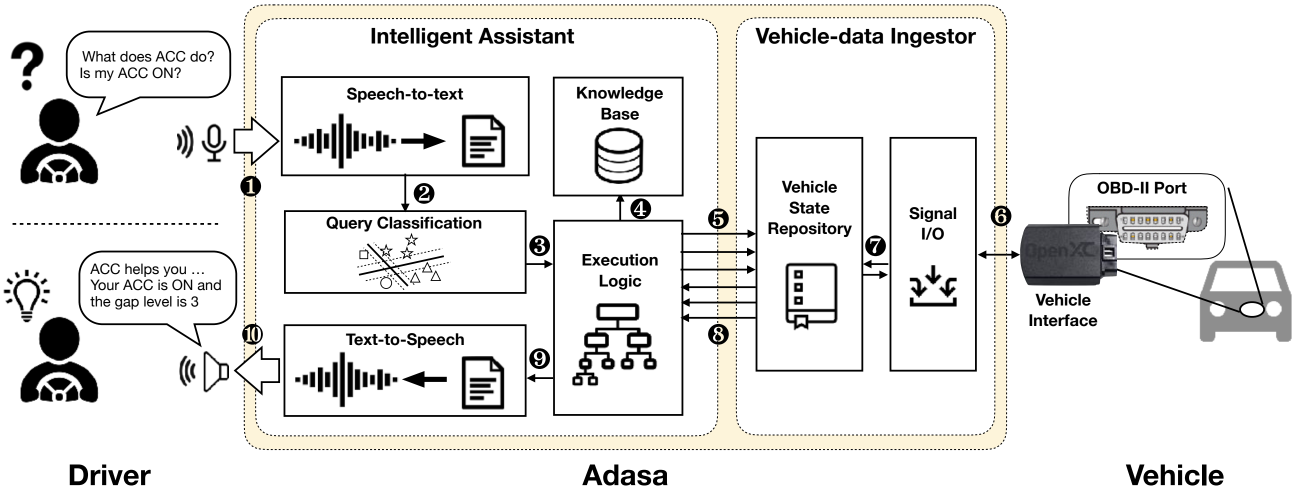

Figure 2: Overview of Adasa system, which is designed based on recent publications [20] and integrated with a commercially available vehicle [8]. Driver's voice captured by the microphone is translated into its text equivalent (❶) and is passed to Query Classifier to identify the type of query (❷). Execution Logic operates the query (❸) and retrieves the information needed from either Knowledge Base (❹) or the vehicle (❺) depends on the type of query. Vehicle-data Ingestor receives the CAN data through the vehicle interface and records the status of each signal in Vehicle State Repository (❻-❼). Once the result is passed back to Execution Logic (❽), the response is converted to a WAV file and output to the speaker to answer driver's questions (❾-❿).

System Overview: The Life of a Query

Adasa provides a voice-enabled user interface for drivers to ask questions about and command and control ADAS features using natural human language, without requiring any complex operations that may distract them from driving. The system leverages contextual information of vehicle state and feature status (e.g., what mode ACC is currently operating in) by accessing the signals from the standard OBD-II port. This allows drivers to ask questions like "*what does the green symbol on my dashboard mean?*" without knowing it is related to ACC. In addition, the system is able to send control signals via OBD-II port to enable or disable features to allow drivers to make commands such as "*can you help me turn on adaptive cruise control?*" instead of knowing or asking about and then pressing the activation buttons on the steering wheel. Leveraging the state-of-the-art framework for building intelligent assistant, Adasa is trained to answer a wide range of questions covering the most frequently asked ones we identified in our analysis on real user verbatims from Ford, and is capable of delivering responsive feedback at real-time.

Figure 2 presents a high-level diagram of the system components and how user queries are handled in Adasa. Adasa is composed of hardware apparatus and software components. On the hardware side, Adasa consists of a compute device and a vehicle interface device. On the software side, Adasa consists of an Intelligent Assistant runtime, and a Vehicle-data Ingestor runtime.

To illustrate how these components are integrated, we walk through the life of a query step-by-step in Figure 2. The life of a query begins with a user's voice input. After the driver activates the system to begin listening by pressing the voice activation button on the steering wheel, the voice input will be sent to the Speech-to-Text engine of Adasa (❶) deployed on the computing device. This voice input is then transcribed into its text equivalent (❷). The text of the query is classified by the Query Classifier, where different intents will be given a different label (❸). For example, questions like "*what does adaptive cruise control do?*" and "*what is adaptive cruise control?*" will be labeled as the same class because they are both querying about the functionality of ACC, but questions such as "*will lane keeping system steer the vehicle for me?*" will be labeled as a different class. The label generated by the Query Classifier will then be sent to the Execution Logic for further analysis. Depending on the intent label of the query, the Execution Logic needs to either fetch the necessary information from a vehicle-specific knowledge base (❹), or sends a request to the Vehicle-data Ingestor to obtain/modify the values of certain vehicle signals(❺). The Vehicle-data Ingestor runs an asynchronous Signal I/O callback which continuously intakes message streams from the vehicle via a Vehicle Interface plugged into the OBD-II port in the vehicle (❻), and maintains a lookup table containing the updated values of all the vehicle signals internally in its Vehicle State Repository (❼). Upon receiving a request from the Execution Logic, Vehicle-data Ingestor searches the current values of the requested signals, updates the signals if the request is a command, and otherwise sends the corresponding signals back to the Execution Logic (❽). Combining all this information, Execution Logic composes a response accordingly in text format, and sends the response to the Text-to-Speech engine (❾) and produces a WAV file. The resulting WAV file of the response is played back to the driver through the in-vehicle speakers (❿), informing drivers with the information they requested.

In the following subsections, we describe each of these components in detail.

Hardware Apparatus

Vehicle

As shown in Figure 1-(B), we employ a 2017 Lincoln Continental Reserve Sedan which is equipped with the Technology Package including ACC. As part of its driver assistance suite, the vehicle is also equipped with LKS, blind spot information system with cross traffic alert, radars and 360 degree camera technology for pre-collision assist and pedestrian detection [8].

OBD-II Port and Vehicle Interface

Our vehicle provides an on-vehicle diagnostic system called On-Board Diagnostic (OBD) system, which gives us access to the internal signals of the vehicle through the Controller Area Network (CAN). OBD-II is an industry standard of such a system implemented in all cars manufactured in the United States from 1996 onward. This standard specifies the pin mapping, the protocol and the message format of the in-vehicle diagnostic connector, and provides engine control, and monitors parts of the body, accessory devices, as well as the diagnostic control network of the car. The interface of OBD-II system (referred as the *OBD-II port* in this work) is usually located within the realm of the driver's seat, underneath the instrument panel or near the footwell. To access the necessary information (e.g., the status of the targeted ADAS features) from the OBD-II port, we plug in a vehicle interface programmed with customized firmware. Vehicle interface is a piece of hardware device that bridges the vehicle and the host device via the OBD-II port. It decodes CAN messages into the software-recognizable format, and sends the decoded results over a common interface, such as USB and Bluetooth, to the host devices. In this work, we use the widely deployed Ford Reference Vehicle Interface (VI), which is a standard open source hardware implementation of the vehicle interface, as shown in Figure 1-(C).

Since the publicly available standard firmware (Type-3 firmware) for a 2017 Lincoln Continental using a Ford Reference VI could not directly access ADAS data, we first reprogrammed and customized the firmware for proprietary access to both LKS and ACC information. An Intrepid Control Systems neoVI is then used to perform additional processing on and writing of CAN signals to allow Adasa to overwrite the vehicle's internal control of ADAS features as part of our prototype. Instead of connecting the OBD VI to our system through a long USB cable in the driver's footwell, which may pose a safety hazard while driving, we enabled a wireless connection between the VI and Adasa via Bluetooth. As shown in the Figure 1-(C), Ford Reference VI plugged into the in-vehicle OBD-II port and two blue LEDs show that VI is successfully recognized by and registered to the vehicle as well as connected to Adasa via Bluetooth.

Vehicle-data Ingestor

We built the Vehicle-data Ingestor by using an open-source OpenXC library to communicate using CAN data between the VI and Adasa with the read and write function enabled. To sustain the consistency of the information between the vehicle and our system, we implemented our Vehicle-data Ingestor in a callback fashion (i.e., where the Vehicle-data Ingestor is able to refresh its own data periodically as the VI receives updates from the vehicle). The stream of CAN data is then analyzed by a sub-procedure in the Vehicle-data Ingestor named Vehicle-State Repository. Vehicle-state repository records the most recent history of the data, extracts updates to each signal from the data, and maintains a lookup table that contains the latest status of each signal. In addition, as Adasa is requested to control and change the status of ADAS features, Vehicle-data Ingestor sends the signal to VI and updates both the lookup table in Vehicle-State Repository, as well as the CAN data inside the vehicle, to ensure the consistency in both sides.

Intelligent Assistant

Intelligent Assistant is designed to understand drivers' questions, process the query coupled with the status of ADAS obtained from Vehicle-data Ingestor, and respond to drivers accordingly. We employ a state-of-the-art speech-based framework, namely Lucida [20], to structure our implementation, including automatic speech recognition and query classification. We describe the details of each component in the following sections.

Speech-to-Text Engine

The first major component in the Intelligent Assistant is a speech-to-text interface. This interface allows drivers to ask the questions in natural language, providing drivers an uninterrupted way to interact with Adasa and access the ADAS features. Adasa builds on the open-source Lucida framework [20], which by default uses Google Speech API [37] for automatic speech recognition (ASR). Despite the noisy on-road environment in our user study, we observed very low error rate, which is similar to prior work reported [39]. It first processes and extracts feature vectors representing the voice segments, and submits the feature vectors to a speech recognition kernel to transcribe drivers' utterances. The transcribed texts then serve as the input to the next engine.

Query Classifier

To access the necessary vehicle information for each query precisely, Adasa needs to first understand the intent of the query. For this we leverage the findings explored in our pilot study, and conclude three types of input queries:

1. **Inquiry (FAQ) -** Queries regarding the explanation of the ADAS features are considered as Inquiry (FAQ). For example, "*How does the lane keeping system work?*", "*What is the gray speedometer on my dashboard?*".

2. **Symptom Diagnosis -** Queries regarding the symptoms or status of the ADAS features are considered as Symptom Diagnosis. For instance, "*Is my lane keeping system active?*", "*I just turned on adaptive cruise control, but why is it not working?*".

3. **Command and Control -** Queries regarding enabling, disabling and changing the status of ADAS features are considered as Command and Control. For example, "*Can you turn on lane keeping system for me?*", "*Please increase the gap distance.*".

One straightforward way to classify queries is to hand-assign a class label for every possible utterance, and use a dictionary-like structure to store the mapping between queries and labels.

During runtime, class label of an input query is determined by applying string matching between the query and the keys in the key space of the dictionary. However, this approach is impractical in real-world driving scenarios since drivers might not have prior knowledge about ADAS and are unfamiliar with the exact terminology. In addition, natural language speech is imprecise and hard to predict. Especially, utterances can significantly deviate from correct grammar during driving since drivers need to focus on the traffic conditions. These observations show that Adasa should be able to handle incomplete and ambiguous sentences which render the hand-coding approach infeasible.

We address these challenges by employing a machine learning based classifier to automate the process of query classification. To encapsulate a large scope of questions for our query classifier, we collect a large amount of training data by using crowd sourcing on *Amazon Mechanical Turk (MTurk)* [4] via the following steps: First, we analyze the customer verbatims and the feedback described in Section 3 to identify the scope of query classes that we have to cover. Second, for each of these classes, we create a task assignment on MTurk, in which we provide a textual description of a driving scenario and a query example to ask under that scenario. We then ask MTurk workers to provide five rephrases of that query. To better contextualize the MTurk workers, we also include a picture of the dashboard of the targeted scenario in the assignment for the workers to gain a better understanding about what they would experience if they were in the vehicle. For example, as shown in the Figure 3, we have assignments in which there are questions such as *"what does it mean if I see a red line and a grey line on the dashboard?"* With the picture of the instrument cluster provided, the workers understand the questions precisely and easily. Third, we collect the completed assignment, manually remove the redundant rephrases, and evaluate the quality of the rest of the rephrases. We include only the qualified rephrases in our final training data. Finally, we train our classifier with this training dataset by utilizing support vector machine (SVM) and deploy the trained classifier in Adasa. We use the unigram and the bigram representations of the input query as features, which is commonly used in text classification. For instance, an input query of "*what is cruise control*" will be transformed into {'what': 1, 'is': 1, 'cruise': 1, 'control': 1, 'what is': 1, 'is cruise': 1, 'cruise control': 1}, where the keys are unigrams and bigrams in the query and the values are their occurrences. We use both unigram and bigram to capture the spatial ordering of words. Overall, over 4500 training data is collected and 73 classes are implemented including 49 Inquiry classes, 11 Symptom Diagnosis classes and 13 Command and Control classes.

Query Classifier outputs a probability distribution for each query, which represents the probability of the query falling into each of the topics. A high probability on one class means the classifier is confident that the query belongs to the corresponding topic. When drivers ask a query that is not covered by the scope the system is trained to understand (e.g., "*how is the weather in San Diego?*"), the output probability distribution will not have any class with a high enough probability (i.e., greater than 0.5). The system identifies such queries as out-of-scope, and respond with "*I am not trained to handle this topic yet. Please ask me about adaptive cruise control and lane keeping systems.*"

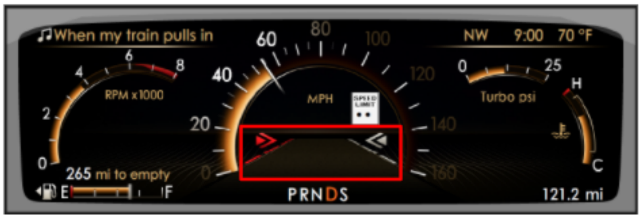

Figure 3: An Amazon MTurk task assignment example. We asked the MTurk workers to rephrase the statement, "*what does it mean if I see a red line and a grey line on the dashboard?*" with the picture of the entire dashboard and a red bounding box around the area of inquiry in order to help the workers understand the questions and task at hand.

Execution Logic

Once the input query is analyzed and classified, Execution Logic then accesses the necessary information and answers the query. Depending on the intent label of the query, Execution Logic fetches information from different sources. To compose a response for *Inquiry (FAQ)* queries, we build a vehicle-specific knowledge base that contains the static information about the vehicle and ADAS features. Adasa accesses the target entry and retrieves the corresponding answer.

To answer *Symptom Diagnosis* queries, on the other hand, Adasa needs to access the states of different features of the vehicle. Adasa, based on the targeted ADAS feature and the intent of the diagnostic query, composes a request and sends it to Vehicle-data Ingestor which decodes the request and seeks for the current status of or a recent update to the targeted signals. Once the requested value is obtained, Vehicle-State Repository returns the result to Execution Logic. We enable non-blocking multi-threaded access from Execution Logic to Vehicle-State Repository, which ensures that Execution Logic can access vehicle states in a timely manner. Finally, Execution Logic applies the returned value and composes a textual response correspondingly.

To complete *Command and Control* queries, Adasa requires to send control signals into the vehicle via VI to alter the ADAS functions. When Execution Logic receives the intent of query from Query Classifier, control signals mapped to the corresponding CAN data messages are generated and transmitted to the VI via Bluetooth and modify the value on the CAN bus inside the vehicle. Also, Execution Logic updates the states of the target signal in Vehicle-State Repository and composes a textual response to indicate that the command is completed.

Text-to-Speech Engine

The output textual response is then translated to speech by the text-to-speech module, where Google Chrome Speech Synthesis library is employed [37]. Several attributes such as the voice of gender, speech rate, pitch and volume can be customized for different users. We use standard Google Female English and set rate, pitch and volume as default. The output speech then is played by the speaker.

EVALUATION

To evaluate our system in improving the perceived usability of ADAS features, we conducted a user study in a real-world driving environment. In this study, each participant interacted with Adasa while driving a 2017 Lincoln Continental Reserve equipped with the Technology Package, which includes ACC and LKS. During the driving study, we focused on ACC and LKS, and encouraged participants to ask Adasa any question, including questions that were beyond the scope of these two ADAS features, so as to obtain complete feedback based on their overall experiences. The study setup is detailed in the following sections.

Participants

We recruited 15 participants: 11 male, 4 female, ages 24—35 (M = 27.1, SD = 3.2). Each participant had a valid US driver's license and was covered by an auto insurance policy sponsored by the university. 80% (12 out of 15) participants did not have prior experience with ADAS, but had sufficient driving experience (M = 8.7, SD = 5.1 driving years). Note that sufficient driving experience is required since participants would be asked to interact with Adasa during the driving study. Participants were recruited through email announcements at the authors' university.

Adasa

As Figure 1-(D) shows, we built Adasa and deployed it in the Lincoln Continental described above to conduct the study. Adasa ran on a laptop placed next to the driver seat and connected through Bluetooth to Ford Reference VI to transmit CAN data via OBD-II port. The left blue light on the VI shows CAN data could be accessed successfully and the right blue light shows the data from the VI was received by Adasa. We prototyped a voice interface with a front-end web application, where participants could ask questions via this interface. To enable Adasa and start the conversation, participants were instructed to press the button on the steering wheel to enable voice recording and send the query to the internal modules of Adasa. The laptop would output the answer using human voice through the speaker in the laptop once Adasa retrieves the results.

The Route

Participants drove on a predefined 11.7 mile (18.8 km) route, which consisted of 9 segments, as Figure 4 shows. The carefully planned route consists of 7 miles (11.3 km) highway and 4.7 miles (7.5 km) suburban road, allowing participants to have enough time and diversity in driving scenarios to accommodate the use of both the LKS and ACC systems. Participants took an average of 20 minutes to complete this route. Participants were asked to complete a different task in every segment, as the next subsection describes.

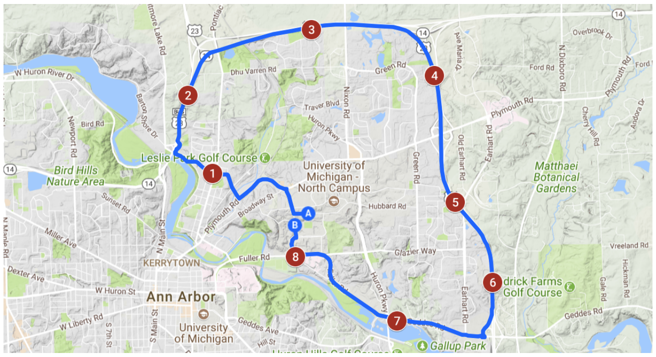

Figure 4: The map of the 11.7 miles route. It includes 7 miles highway and 4.7 miles suburban road to accommodate the use of both ACC and LKS.

Table 1: Summary of tasks assigned during the driving study.

Segment #	Task
0	**Start driving and be familiar with the vehicle**
1	**Turn on the lane keeping system**
2	**Get on highway and stay in right-most lane**
3	**Turn the adaptive cruise control to standby**
4	**Turn the adaptive cruise control to active**
5	**Drift out of lane slightly to drive on the lane line**
6	**Turn off the adaptive cruise control**
7	**Drift out of lane slightly to drive on the lane line**
8	**Turn off the lane keeping system**

Tasks

The detailed tasks are shown in the Table 1. The first segment (segment 0) is designed for the participants to become familiar with operating the vehicle and asking Adasa questions. Segment 1 consists of another portion of the local road including several stop signs where the participants were asked to turn on LKS. In this segment, participants continued driving while LKS is enabled on the local road. The participants are then asked to get on the highway after finishing segment 1 and stay in the rightmost lane while keeping the vehicle speed at 60 miles per hour for safety. After entering the highway, the participants were asked to turn ACC to standby mode and active mode in segment 3 and segment 4 respectively. These two segments were designed for the participants to experience the ACC feature and ask Adasa questions if they chose to do so. In the segment 5, the participants were asked to drift out of lane slightly at the right-most lane to experience the LKS feature while ensuring that the vehicle is under control and the speed limit is followed. In segment 6, the participants were asked to get off the highway, turn off the ACC, and drive back to the starting location. Segment 7 was designed to let the participants experience LKS in the local area. This segment is a straight urban road with only one traffic light. During this segment, the participants were asked to test LKS again by drifting out slightly to step on the lane line. Finally, participants were asked to turn off LKS and drive to the end location. We expected drivers would encounter most of the driving scenarios when executing tasks during the drive, although this is not always possible. For example, even when ACC is set to active, it can still be hard to expect the driver to experience the Stop-and-Go function (i.e., detect when the vehicle ahead has stopped, and resume after the vehicle ahead moves) since the traffic conditions on road may vary.

Procedure

Upon the participant's arrival at the starting location, the participant was greeted and told that his/her help was needed to investigate the driving experience on a vehicle equipped with ADAS features and that he/she would interact with our digital assistant, Adasa. Also, each participant was required to sign an insurance form to obtain the university's approval for participating in the study. After asking for the participant's informed consent and checking his/her valid driver license, the participant was introduced to the vehicle and asked to "please have a seat in the drivers' seat". After both the researcher and participants were seated, the researcher showed the participant the pre-defined route. The participant was then informed that varying tasks were needed to be performed during each segment and was encouraged to ask any questions. These questions were not limited to questions about the two ADAS features, as long as the participants felt they were helpful for resolving their confusion. Since most of the participants were unfamiliar with ADAS features, which may share vehicle control with the driver, the researcher instructed the participant to prioritize their safety and allowed them to interrupt the study at any time if they felt nervous or were in a dangerous situation. The researcher stayed in the vehicle to guide the participant along the route as well as the tasks needed to be executed along each segment. As soon as the participant understood all the instructions, he/she was allowed to start the study and was instructed to utilize the first segment (segment 0) to get familiar with driving the vehicle.

Upon finishing the route and the arrival at the end location, the participant was asked to fill out a post-questionnaire regarding his/her driving experience during the drive. Afterwards, the participant was interviewed by the experimenter for more detailed information, and participants were asked about any further questions regarding the vehicle, Adasa, or ADAS features. At the point where the participant expressed he/she had no further questions, the study was deemed formally over.

Questionnaire

Participants filled out a questionnaire adapted from DeLone and McLean information-system (IS) success model [10, 11], with questions that evaluate three main aspects of the system: system quality, information quality and user satisfaction. All questions in the questionnaire are specifically asked regarding the assistant itself without considering the quality of the ADAS features. Therefore, the evaluation is scoped to evaluate the effectiveness of Adasa in improving drivers' understanding of ADAS features rather than the usability of ADAS features on the vehicle. The participants answer each question in the categories with a 10-point Likert scale with anchors 1 = "strongly disagree" and 10 = "strongly agree". For system quality, the participants are asked to evaluate system acceptability, system intelligence and system helpfulness. For example, "*The digital assistant helps me understand the features of advanced driver assistance system.*" For information quality, the information usefulness and the accuracy of the diagnosis are evaluated. One example question is: "*I think the digital assistant can understand my question and provide accurate diagnosis during driving.*" For the user satisfaction, the participants are asked regarding their nervousness, pleasantness and if their expectation had been met. The targeted assessments of each question in the questionnaire are shown in Table 2. The participants were also welcome to provide comments at the end of the questionnaire for us to evaluate the system.

RESULTS

With the user study conducted in a real-world environment, we are able to investigate drivers' behaviors and satisfaction when interacting with Adasa. In this section, we demonstrate the effectiveness of our system via quantitative and qualitative analyses of our user study.

Quantitative System Analysis

We perform a quantitative system analysis by evaluating three key metrics: (1) query understanding, (2) response correctness, and (3) processing latency. Particularly, query understanding (i.e., understanding the query correctly) and response correctness (i.e., responding with the correct answer) have been commonly used to evaluate conversational assistants such as Apple Siri and Amazon Echo [32, 33].

Query Understanding

We first evaluate query understanding by quantifying the percentage of the queries that are categorized into the corresponding intent class correctly, which aligns with how query understanding has commonly been evaluated in prior studies [32, 33] on several digital assistants including Amazon Echo, Google Home and Apple Siri. The experiment including 800 general queries was conducted in April 2017 for Apple Siri and August 2017 for Amazon Echo and Google Home respectively. To facilitate a fair comparison, we divide the data we collected via Amazon MTurk into two completely disjoint sets, the training set and the testing set, and report the accuracy of our model on the testing set to provide an unbiased evaluation. As shown in prior studies [32, 33], state-of-the-art digital assistants (e.g., Siri) are able to identify and understand over 90% of the queries asked. In our evaluation, we find Adasa is also able to categorize queries at an accuracy of 92.5%, which suggests that our system can achieve state-of-the-art query understanding.

Response Correctness

We then evaluate the response correctness, which can be quantified as the percentage of the queries answered correctly by the digital assistant. For Adasa, we measure the response correctness during the user study, where the participants were asked to inform the experimenter whether Adasa provided the correct answers or not during the test drive. The results in our user study show that each participant asked 12.88 questions on average (i.e., 1–3 questions per segment) during the study. We found that Adasa achieves overall 77% response correctness, which aligns with the subjective user feedback that we present later this section. Overall, we found the participants highly satisfied with responses provided by the system. Comparing to the state-of-the-art digital assistants available on the market, our system achieves a comparable, if not better (i.e., 75.4% correctness on Apple Siri, 65.3% on Google Home, and 53.6% on Amazon Echo as shown in prior studies [32, 33]), level of response correctness.

Table 2: Targeted assessments of questions in questionnaire.

Question #	Description
Q1	**I think it is acceptable to have a voice assistant in the vehicle.**
Q2	**I think this car equipped with the voice assistant is intelligent.**
Q3	**The voice assistant helps me understand the features of advanced driving assistant system.**
Q4	**I think the answer that provided by the voice assistant is useful.**
Q5	**I think the voice assistant can understand my question and provide accurate diagnosis during driving.**
Q6	**The voice assistant makes using these advanced driving assistance features more pleasant.**
Q7	**Driving the vehicle with the voice assistant makes me nervous.**
Q8	**The responsiveness and reliability of the voice assistant meet my expectation.**

Processing Latency

Besides the response correctness, the performance of the system can also be determined by the processing latency. A well-designed digital assistant should be able to process queries and respond in real-time, which is also the second key design objective mentioned in Section 4.1. We measure the processing latency with respect to query types (i.e., inquiry, symptom diagnosis and command and control) in the real vehicle deployment. The result demonstrates that it merely takes 1.50 seconds (M = 1.50, SD = 0.38) from driver pressing the button to Adasa responding on average. Specifically, the processing latency across three types of query is 1.17 seconds for FAQ (M = 1.17, SD = 0.19), 1.57 seconds for diagnosis (M = 1.57, SD = 0.21) and 1.76 seconds for command (M = 1.76, SD = 0.45).

Subjective User Feedback

Figure 5 presents the average scores of all the eight questions, where the x-axis represents different questions regarding Adasa usage with the average scores of these questions shown on the y-axis. A score of 7 out of 10 or greater was seen in all cases except 'nervousness', which will be discussed later, suggesting that Adasa is considered to be a helpful and useful system for them to understand and utilize ADAS features in the vehicle, which aligns with the third design objective. An interesting observation is that 80% participants (12 out of 15) in our study have not used ADAS features previously since their own vehicles are not equipped with ADAS features, or they are unfamiliar with them. Adasa improves the understanding of ADAS as participants provide highly positive feedback (Q3 score = 8.9), showing Adasa is helpful to understand these features.

System Quality

System quality was evaluated by asking participants questions about the following three aspects: system acceptability, system intelligence, and system helpfulness. As shown in Figure 5, participants considered that an in-vehicle digital assistant like Adasa "highly acceptable" for in-vehicle use (Q1 score = 9.3) and it could help them understand the features of ADAS (Q3 score = 8.9). Participants mentioned: "*The voice assistant is convenient and makes me drive more safely as I could focus on the road all the time when driving and get responses I needed*", "*It is useful to have while driving since I can keep my eyes on driving instead of seeing the dashboard.*", "*The voice assistant makes it much easier to access the ADAS features that are complex in the manual. Using voice is a much more natural way to interact with intelligent features in the car.*"

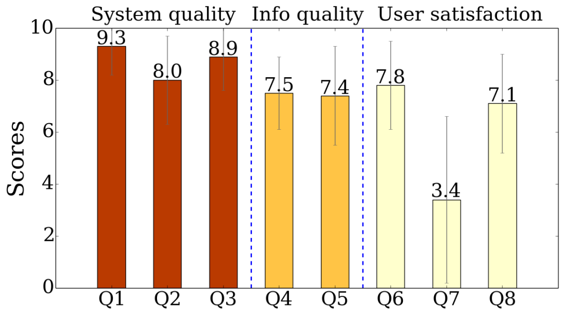

Figure 5: The average scores of the participants' feedback across different questions in the questionnaire. We obtain over 7 out of 10 in all cases and the participants are less nervous when using Adasa (Q7 score = 3.4). It suggests that Adasa is considered as a helpful and useful system for them to understand and utilize ADAS features in the vehicle.

Information Quality

When evaluating the information quality Adasa provided, the expectation regarding the usefulness and the understandability of the information provided by Adasa are considered. Based on the feedback, most of the responses are precise and easy-to-understand (Q4 score = 7.5; Q5 score = 7.4). However, we observe that participants frequently asked questions regarding events that happened a short time ago. For example, one question was asked: "*What happened 3 minutes ago that my adaptive cruise control did not work at all?*" While Adasa currently is able to only respond to the queries based on the current status, this significant finding would help us improve our prototype in the future.

User Satisfaction

We evaluate the user satisfaction by considering participants' pleasantness, nervousness, and if Adasa could meet their expectations during driving. The results show that participants feel pleasant (Q6 score = 7.8) and Adasa could meet their expectations (Q8 score = 7.1), as shown in Figure 5. For user nervousness, participants indicated a score between 1 (not feeling nervous at all) and 10 (very nervous). Based on the results, while the average level of nervousness is low among all the participants (Q7 score = 3.4), we find two participants provided a score of 8 and a score of 9 respectively, showing they were highly nervous during driving. They mentioned that they remained conservative about using these features even though they considered Adasa helpful for them to understand these features. Apart from these two participants, other participants gave mostly 1's and 2's, showing they did not feel very nervous using Adasa while driving.

DISCUSSION

While our results indicate that an in-vehicle digital assistant comparable to Adasa can indeed improve ADAS feature usability and the overall perceived driving experience, we also identified numerous practical insights and challenges. Particularly, there are two common themes arising from our observations: (1) response length; and (2) query completeness.

Response Length

Participants are able to interact with Adasa in several scenarios (e.g., driving as in on the highway or statically as in a parking lot). We find that participants react differently to similar responses in the different environment. For example, participants often anticipate brief responses while driving in a dynamic environment (e.g., highway, road intersection) since they are unable to digest the information provided by Adasa and focus on the road environments simultaneously. In contrast, more thorough explanations are expected as participants interact with Adasa in a static environment (e.g., parking lot) to understand how to use these ADAS features. This observation aligns with our feedback from the questionnaire that 60% (i.e., 9 out of 15) of the participants commented that they are satisfied with the comprehensive information provided and are able to readily understand and more quickly familiarize themselves with those particular ADAS features. Consequently, an Adasa-like system should be able to identify the current driving status and provide proper length of responses accordingly since different response lengths are expected depending on driver's current status.

Finding - *Different length of responses are expected in varying driving environments as drivers might be distracted while driving. A well-designed digital assistant is capable of identifying the environment and providing proper length of responses.*

Query Completeness

We find that questions asked by drivers are usually incomplete and unstructured because of the following reasons: (1) drivers are unfamiliar with ADAS features so they are often unable to describe their questions precisely; (2) drivers often pay attention to the traffic conditions while interacting with Adasa-like system, which makes it difficult for them to structure complete sentences. However, Adasa is trained with the training data collected by MTurk, which is mostly comprised of complete sentences since those workers were unable to experience the system while driving and respond as such. As a participant mentioned: *"Most of the questions that I had the system could answer, but I had to repeat myself multiple times."* This demonstrates that drivers focus mostly on the road and their utterances might significantly deviate from correct grammar or complete sentences. Although our system can achieve up to 77% response correctness, this observation shows that training data for the classifier should include more diverse queries to build a much robust classifier for an Adasa-like system.

Finding - *Incomplete queries are asked frequently since drivers need to pay attention to traffic conditions and might not have prior knowledge to describe the questions precisely.*

CONCLUSION AND FUTURE WORK

We present Adasa, a conversational in-vehicle digital assistant that intakes and answers driver's questions in natural spoken language. Drivers are able to ask questions about or command and control both ACC and LKS using unconstrained natural language in the vehicle. The digital assistant trained using advanced machine learning techniques coupled with access to the vehicle signals responded in real-time based on conversational and environmental context. Results of the system deployed onto a production vehicle were presented demonstrating its effectiveness in improving driver understanding and usability of the ADAS.

We envision a few directions for future work. We believe that an Adasa-like system could educate drivers about ADAS features and broaden their knowledges about ADAS. Future studies could also evaluate longer-term learning gains by utilizing Adasa and the impacts to the driving experience. In addition, we anticipate gathering more variation in training data (e.g., incomplete queries) to further build a robust in-vehicle digital assistant based on our feedback observation. Last, the ADAS features presented in this paper only constitutes LKS and ACC. Extensions of this work would consider other features such as forward collision warning (FCW) and automatic parallel parking for users to understand and utilize these features and to further improve the driving experience.

We believe that in the future more ADAS features will continue to be introduced to improve the driving experience and drivers will be able to utilize an Adasa-like system to interact with the vehicle. We hope Adasa will encourage discussion and excite more in-vehicle interface design within the HCI community.

ACKNOWLEDGEMENTS

We thank our anonymous reviewers for their feedback and suggestions. We would also like to thank Clinc engineers who have provided valuable feedback to this work. This work was sponsored by Ford Motor Company.

REFERENCES

1. Apple. 2017. iOS-CarPlay-Apple. (2017).
2. Adriana Barón and Paul Green. *Safety and usability of speech interfaces for in-vehicle tasks while driving: A brief literature review*. Technical Report.
3. Michael Braun, Nora Broy, Bastian Pfleging, and Florian Alt. 2017. A design space for conversational in-vehicle information systems. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 79.
4. Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. 2011. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science* 6, 1 (2011), 3–5.
5. Seunghyuk Choi, Florian Thalmayr, Dominik Wee, and Florian Weig. 2016. Advanced Driver-Assistance Systems: Challenges and Opportunities Ahead. (2016). McKinsey&Company.
6. Ford Motor Company. 2017a. Ford Sync. (2017).

7. General Motors Company. 2017b. Chevrolet MyLink: Take Control Of Your Vehicle's Technology. (2017).

8. Lincoln Motor Company. 2017c. 2017 LINCOLN CONTINENTAL. (2017). https://media.lincoln.com/content/dam/lincolnmedia/lna/us/product/2017/continental/Lincoln-Continental-Product-Card.pdf

9. Toyota Motor Corporation. 2017. Toyota-Entune. (2017).

10. William H DeLone and Ephraim R McLean. 1992. Information systems success: The quest for the dependent variable. *Information systems research* 3, 1 (1992), 60–95.

11. William H Delone and Ephraim R McLean. 2003. The DeLone and McLean model of information systems success: a ten-year update. *Journal of management information systems* 19, 4 (2003), 9–30.

12. Alfred Eckert, Andree Hohm, and Stefan Lueke. 2013. An integrated ADAS solution for pedestrian collision avoidance. In *Proceedings of the 23rd International Conference on the Enhanced Safety of Vehicles, Seoul, Republic of Korea*. 13–0298.

13. McKinsey Center for Future Mobility. 2017. Autonomous Driving. (2017).

14. Yannick Forster, Frederik Naujoks, and Alexandra Neukum. 2017. Increasing anthropomorphism and trust in automated driving functions by adding speech output. In *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 365–372.

15. Magdalena Gärtner, Alexander Meschtscherjakov, Bernhard Maurer, David Wilfinger, and Manfred Tscheligi. 2014. Dad, stop crashing my car!: Making use of probing to inspire the design of future in-car interfaces. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 1–8.

16. Google. 2017. Android Auto. (2017).

17. Robert Graham and Chris Carter. 2000. Comparison of speech input and manual control of in-car devices while on the move. *Personal Technologies* 4, 2 (2000), 155–164.

18. Boston Consulting Group. 2015. Nearly 10,000 Deaths Could Be Prevented and More Than $250 Billion Saved with Greater Use of Driver Assistance Technologies. (2015).

19. John HL Hansen, Pongtep Angkititrakul, Jay P Plucienkowski, Stephen Gallant, Umit H Yapanel, Bryan L Pellom, Wayne H Ward, and Ronald A Cole. 2001. " CU-move": analysis & corpus development for interactive in-vehicle speech systems.. In *INTERSPEECH*. 2023–2026.

20. Johann Hauswald, Michael A Laurenzano, Yunqi Zhang, Cheng Li, Austin Rovinski, Arjun Khurana, Ronald G Dreslinski, Trevor Mudge, Vinicius Petrucci, Lingjia Tang, and others. 2015. Sirius: An open end-to-end voice and vision personal assistant and its implications for future warehouse scale computers. In *ACM SIGPLAN Notices*, Vol. 50. ACM, 223–238.

21. Malte F Jung, David Sirkin, Turgut M Gür, and Martin Steinert. 2015. Displayed uncertainty improves driving experience and behavior: The case of range anxiety in an electric car. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2201–2210.

22. Dagmar Kern and Albrecht Schmidt. 2009. Design space for driver-based automotive user interfaces. In *Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 3–10.

23. Philip Koopman and Michael Wagner. 2016. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety* 4, 1 (2016), 15–24.

24. Philip Kortum. 2008. *HCI beyond the GUI: Design for haptic, speech, olfactory, and other nontraditional interfaces*. Morgan Kaufmann.

25. David M Krum, Jens Faenger, Brian Lathrop, Jo Ann Sison, and Annie Lien. 2008. All roads lead to CHI: interaction in the automobile. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2387–2390.

26. John D Lee, Brent Caven, Steven Haake, and Timothy L Brown. 2001. Speech-based interaction with in-vehicle computers: The effect of speech-based e-mail on drivers' attention to the roadway. *Human factors* 43, 4 (2001), 631–640.

27. Key Jung Lee, Yeon Kyoung Joo, and Clifford Nass. Partially Intelligent Automobiles and Driving Experience at the Moment of System Transition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*.

28. Victor Ei-Wen Lo and Paul A Green. 2013. Development and evaluation of automotive speech interfaces: useful information from the human factors and the related literature. *International Journal of Vehicular Technology* 2013 (2013).

29. Clifford Nass, Ing-Marie Jonsson, Helen Harris, Ben Reaves, Jack Endo, Scott Brave, and Leila Takayama. 2005. Improving automotive safety by pairing driver emotion and car voice emotion. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1973–1976.

30. Don Norman. 2013. *The design of everyday things: Revised and expanded edition*. Basic Books (AZ).

31. Eshed Ohn-Bar and Mohan Manubhai Trivedi. 2014. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems* 15, 6 (2014), 2368–2377.

32. Loup Ventures Research. 2017. Faceoff: Amazon echo show VS Google home part II. (2017).

33. Loup Ventures Research. 2018. Siri semester exam grade improves to C from D+. (2018).

34. Albrecht Schmidt, Anind K Dey, Andrew L Kun, and Wolfgang Spiessl. 2010. Automotive user interfaces: human computer interaction in the car. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. ACM, 3177–3180.

35. David Sirkin, Nikolas Martelaro, Mishel Johns, and Wendy Ju. 2017. Toward Measurement of Situation Awareness in Autonomous Vehicles. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 405–415.

36. David L Strayer, Joel M Cooper, Jonna Turrill, James R Coleman, and Rachel J Hopman. 2017. The smartphone and the driverâĂŹs cognitive workload: A comparison of Apple, Google, and MicrosoftâĂŹs intelligent personal assistants. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale* 71, 2 (2017), 93.

37. Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics* 1 (2013), 1–12.

38. Sergej Truschin, Michael Schermann, Suparna Goswami, and Helmut Krcmar. 2014. Designing interfaces for multiple-goal environments: Experimental insights from in-vehicle speech interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 1 (2014), 7.

39. Venturebeat. 2017. Google's speech recognition technology now has a 4.9% word error rate. (2017).

40. Fuliang Weng, Pongtep Angkititrakul, Elizabeth E Shriberg, Larry Heck, Stanley Peters, and John HL Hansen. 2016. Conversational In-Vehicle Dialog Systems: The past, present, and future. *IEEE Signal Processing Magazine* 33, 6 (2016), 49–60.

41. Fuliang Weng, Sebastian Varges, Badri Raghunathan, Florin Ratiu, Heather Pon-Barry, Brian Lathrop, Qi Zhang, Harry Bratt, Tobias Scheideck, Kui Xu, and others. 2006. CHAT: A conversational helper for automotive tasks. In *Ninth International Conference on Spoken Language Processing*.

42. Xenophon Zabulis, Haris Baltzakis, and Antonis A Argyros. 2009. Vision-Based Hand Gesture Recognition for Human-Computer Interaction. *The universal access handbook* 34 (2009), 30.