SPEECH-ENABLED INFORMATION RETRIEVAL IN THE AUTOMOBILE ENVIRONMENT

Yeshwant Muthusamy, Rajeev Agarwal, Yifan Gong and Vishu Viswanathan

DSP Solutions R&D Center
Texas Instruments Incorporated
P.O. Box 655303, MS 8374, Dallas, TX 75265

ABSTRACT

With recent advances in speech recognition and wireless communications, the possibilities for information access in the automobile have expanded significantly. In this paper, we describe four system prototypes for (i) voice-dialing, (ii) Internet information retrieval—called InfoPhone, (iii) voice e-mail, and (iv) car navigation. These systems are designed primarily for hands-busy, eyes-busy conditions, use speaker-independent speech recognizers, and can be used with a restricted display or no display at all. The voice-dialing prototype incorporates our hands-free speech recognition engine that is very robust in noisy car environments (1% WER and 3% string error rate on the continuous digit recognition task at 0 db SNR). The InfoPhone, voice e-mail, and car navigation prototypes use a client-server architecture with the client designed to be resident on a phone or other hand-held device.

1. INTRODUCTION

We describe our work toward voice access to multiple information services within the automobile. In addition to the ability to dial phone numbers by speaking the digits, other examples of such services include name-based dialing, Internet information retrieval (InfoPhone), accessing and sending e-mail by voice, and car navigation. Common to all of these services are the issues of noise robustness, limited resources, accuracy, and naturalness of the system.

Noise Robustness. The noise environment of the automobile represents a central concern for recognition accuracy. Recent advances in microphone design that also apply to the car include close talking, active noise canceling microphones and highly directional array microphones. Such solutions provide the best speech signal for recognition, but might raise objections based on aesthetics or cost. In the section on voice dialing, we investigate noise robustness techniques that work well down to an SNR of 0 dB.

Client-Server Architecture. We expect the front-end processing of the speech (waveform to some spectral representation) to be handled by DSPs in the client device(s). Further, we assume that the client will include sufficient memory and CPU power at a reasonable cost for small to moderately large vocabulary applications. In some cases it might be preferable to offload larger vocabulary tasks to a recognition server; we discuss this possibility in the context of the voice e-mail and car navigation systems.

Speech Recognition. To better support recognition tasks within the automobile, we have developed a speaker- independent, continuous speech recognition system called DAGGER (Directed Acyclic Graphs of Grammars for Enhanced Recognition) that has dynamic grammar creation and switching capability [1, 2]. This means that the system can work in new contexts with new vocabularies and without training. This approach supports increased robustness and lessens resource requirements by limiting the context to exactly that needed at the time. We use this approach for the InfoPhone, voice e-mail, and car navigation prototypes.

Java Speech API. We have developed a Speech API [2] for Java [3] on top of DAGGER. This API is a subset of the official Sun JSAPI [4]. Our API allows us to speech-enable Java applets and applications with little effort. Further, Java has several network-friendly features like downloadable applets, remote method invocation and object serializability, which render it useful for client-server wireless phone/device applications. For example, some of these applications need not reside on the device at all, but can be downloaded on demand and removed after use, freeing up valuable memory on the device for other tasks. The InfoPhone, voice e-mail, and car navigation prototypes are implemented in Java.

Speech Synthesizer. A key requirement of displayless user interfaces (such as those found in an automobile environment) is the provision of audio feedback to the user, about what the system is doing. We use a text-to-speech (TTS) system to provide the speech playback in the InfoPhone, voice e-mail, and car navigation systems. Most TTS systems tend to sound robotic, and can tire users during a long session. However, pre-recorded prompts and messages are an expensive alternative and not flexible enough to handle all scenarios. We chose the L&H/Centigram TruVoice

TTS system as it had the right combination of reasonable voice quality and ease of integration (API available). We wrote a Java wrapper class for this TTS API. The Java class exposes all of the API functions and allows us to easily add speech synthesis to any Java applet or application.

Natural User Interface. For most applications in the automobile, the system should work in "hands-busy, eyes-busy" conditions. While the TTS system provides a good output interface in such conditions, the input interface also needs to be such that the user can talk to the system in a natural manner rather than having to remember cryptic commands. This often leads to the user making incorrect, incomplete, ambiguous, or inconsistent requests to the system. Therefore, it is important to have a dialog manager that interacts with the user to resolve these requests and ensures that the user obtains the desired information. The ultimate objective is to make the user's interaction with this system as similar to human interaction as possible. We explore these issues in the InfoPhone, voice e-mail and car navigation prototypes.

The following sections describe the four "systems" in greater detail. It is important to note that these systems are not products but prototypes that demonstrate our technology for in-car consumer devices of the near future.

2. VOICE DIALING

Voice dialing provides spoken name dialing, digit (phone number) recognition, and simple voice commands. The key issue involved in voice dialing is the robustness of the speech recognizer. A speech recognizer used in an automobile must be robust to acoustic variations in the speech signal which do not carry linguistic information. Such acoustic variations, which may cause severe performance degradation, may be caused by: (i) noises from the engine, A/C, outside air flow, and tires, especially in highway conditions, (ii) speaker dialect and accent, and (iii) microphone frequency response, mounting positions (hand-held/hands-free), and the speaker's position relative to the microphone. The acoustic environment changes as a function of driving conditions.

2.1. Adaptation Techniques

Except for name dialing, which is speaker-dependent and thus trained to a specific user, a set of speaker-independent speech models must be initially provided. However, direct use of these models in the car without any adaptation will result in substantial loss of performance due to the above-mentioned acoustic variations. We have developed two adaptation techniques to maintain recognition performance. Our first adaptation technique deals with environment or background noise. With some noise data sampled in the car, the speech models are modified to accommodate noisy speech.

Table 1: **Highway noise error reduction with automatic noise adaptation and 10 utterances for speaker adaptation (baseline of 0.35% WER on clean speech).**

Adaptation	*% WER*
None	60
Noise	4
Noise & Speaker	1

This adaptation occurs automatically, in an *unsupervised* manner, for each utterance allowing for dynamically changing background noise. Our second adaptation technique, which deals with speaker and microphone variability [5], involves *supervised* one-time only adaptation, and is carried out before first use of the system. Based on a few utterances recorded in the car with the engine off, the speech models are transformed to match the speaker and the interior acoustic environment of the car, including the microphone characteristics. We have the ability to adapt based on any number of utterances containing a variety of words. Although more utterances always result in lower word error rates, we find that as few as 5 to 10 utterances provide sufficient adaptation for continuous digit recognition.

2.2. Performance

The two adaptation techniques are combined to give improved accuracy and robustness to noise in the car environment. To test the robustness of our recognizer, we tested these techniques on a continuous digit recognition task containing one- to seven-digit utterances (from the TIDIGITS corpus). The noise level was scaled to produce an SNR of 0 dB. For each test utterance, we automatically adapted the models for noise by using approximately the first 0.25 seconds of noise before detection of speech. To adapt to the speaker, we ran adaptation on ten seven-digit utterances chosen at random from the test set and tested on the remaining utterances. Table 1 shows the results where the recognition performance is measured by %WER (percent word error rate). These performance improvements in extremely harsh conditions approach the error rate of 0.35% achieved in the original quiet office environment conditions.

3. INFOPHONE

Although the mobile user may have access, via radio broadcasts, to useful information such as stock quotes, flight schedules and weather forecasts, this information is (i) not customized to each user, and (ii) only available when he is in the car or has other access to a radio.

Our InfoPhone prototype system attempts to solve this problem by providing easy access to information sources on the web from the user's cellular phone. The InfoPhone is currently a speech-enabled Java applet that simulates a cellular phone. Users can choose one of flights, stocks and weather from a top level menu and interact with each "service" by speech commands. "Keypad" (non-speech) input is also available as a backup. The applet incorporates separate grammars for company names (for stocks), flight numbers and city names (for weather). We envision these grammars to be customized for each user when he signs up for the system. The dynamic grammar switching capability of our DAGGER recognizer allows the user to switch between these grammars on-the-fly. Speech input to the applet is processed by the recognizer and the information request is sent to a server that accesses the appropriate website, retrieves the HTML page, extracts just the essential information and transmits it to the applet. The results of the information retrieval are played out by the TTS system and displayed on the small "phone display". This way, the user is not forced to look at the display for the information. We expect the InfoPhone system to be a valuable information retrieval tool for the mobile user.

4. VOICE E-MAIL

Over the past several years, the cellular telephone has become an important mobile communication tool. The use of voice mail has also increased over the same time period. At the same, communication by e-mail, once limited to government research organizations, universities and high-tech companies, has now become commonplace. The convenience, speed and ease-of-use offered by e-mail has made it a viable, often preferred, alternative to other more traditional forms of communication, such as paper mail and telephones. Therefore, it would be convenient if the mobile user could use a single device (such as a cellular phone) to access both his e-mail and voice-mail. This eliminates the hassle of dealing with multiple devices and also allows multimodal messaging; a user can call up the sender of an e-mail message to respond verbally to his e-mail, or send e-mail in response to a voice-mail message.

4.1. System Overview

The Voice E-mail (VE) system has a client-server architecture and is completely voice-driven. Users talk to the system and listen to messages and prompts played back by the speech synthesizer. The system has a minimal display (for status messages) and is designed to operate primarily in a "displayless" mode, where the user can effectively interact with the system without looking at a display. The current system is an extension of previous collaborative work with MIT [6] and handles reading, filtering, categorization and navigation of e-mail messages. It will soon incorporate voice-mail send and receive (using Caller ID information) and later, the capability to "compose" and send e-mail (for example, using speech-based form-filling).

An important aspect of the displayless user interface is that the user should, at all times, know exactly what to do, or should be able to find out easily. To this end, we have incorporated an elaborate context-dependent help feature. If the user gets lost, he also has the ability to reset all changes and start over from the beginning. For people who prefer a display for non-driving conditions, an optional display can be incorporated into the VE system.

4.2. Client-Server Architecture

The server handles all of the e-mail/voice-mail functions. It accesses the e-mail and voice-mail servers and handles the receiving, sending and storage of the messages. It communicates with the client via sockets. The server also handles parts of the TTS system and the speech recognition for sending messages. The server is implemented as a Java application. The client provides the user interface and handles the reading, navigation, categorization, and filtering of e-mail and voice-mail messages. It has both speech recognition and TTS capabilities and does not maintain constant connection to the server (to reduce connection time charges). It connects to the server only to initiate or end a session, check for new mail or to send a message. It also has an extensive help feature that provides guidance to beginners of the system and on request. The client is implemented as a Java applet.

4.3. User Interface

The user can speak to the system in a natural, continuous speaking style. Several alternates to each phrase are allowed (for example, "any messages from John Smith?" and "is there a message from John Smith?"). There is also a rejection feature that handles incorrect speech input. It prompts the user for more information if the recognition score falls below an empirically determined threshold. To minimize fatigue, the error prompts in case of a rejection are randomized. Further, if more than three consecutive rejections occur, the system drops into context-dependent help to guide the user. The TTS system operates in e-mail mode; that is, it can correctly speak out the e-mail headers.

5. CAR NAVIGATION BY VOICE

Car navigation systems have been available for some time, but they have received only limited use. We can partly attribute this to the user interface available for such systems: often unnatural, sometimes clumsy, and potentially unsafe.

Some systems use a touch screen while others use a rotating knob to enter destination addresses one alphanumeric character at a time. We have developed a system to obtain maps and/or directions for different places in a city as naturally as possible, by voice I/O only. It could be incorporated into either a built-in device in a car or a cellular phone. This navigation system is primarily aimed at hands-busy, eyes-busy conditions such as automobile driving. An optional display is provided for situations where the user may safely look at the screen, like when the car is parked. All textual information is played back to the user via a TTS system. A dialog manager is used to handle all interactions with the user.

5.1. Client-Server Architecture

The car navigation device acts as a client. The user interacts with the client which in turn communicates with a remote server to process user utterances. A Global Positioning System (GPS) installed on the client tracks the location of the user at any point in time. A web-based map service on the server provides maps and directions. We currently use the $MapQuest^{TM}$ web site as our map server (www.mapquest.com). Further, a yellow pages server is used to find businesses near the user's current location. We use the $GTESuperPages^{TM}$ web site as our yellow pages server (www.superpages.com). Our DAGGER recognizer processes the user's utterances and passes the result to the dialog manager, which then interprets these utterances in context. If the appropriate information needed to issue a query has been given, the dialog manager will query the appropriate server to get a response. Otherwise, it may interact further with the user. For example, if the user says *"Where is the DoubleTree Hotel?"* and the system has knowledge of multiple hotels of the same name, it will first interact with the user to resolve this ambiguity before querying the map server.

The navigation application has been designed so that the user may query the system using natural speech. The speech interface provides a natural way for users to specify the destination, while the presence of a dialog manager ensures that users can have their queries satisfied even in the presence of missing, ambiguous, inconsistent, or erroneous information. The dialog manager also assists in constraining the grammars for the speech recognizer and in providing context-sensitive help. This dialog manager has been described in greater detail elsewhere [7]. In case of any errors on the part of the user or the system, the user may say *"Go Back"* at any time to undo the effect of the previous utterance. It also supports a rejection feature that requests the user to repeat something if the system does not have enough confidence in its recognition.

5.2. Navigation Scenarios

This application covers different scenarios in which a user may need directions to some place. In some cases, the user may know the exact address or cross streets of the destination and might query the system for directions to these locations (for example, *"How do I get to 8330 LBJ Freeway in Dallas?"*). In addition, the system has knowledge of a list of common points of interest for the current city. These may include hotels, hospitals, airports, malls, universities, sports arenas, etc., and the user can get directions to any of these by referring to them by name (for example, *"I need to go to the Dallas Museum of Art"*). Finally, there are often instances where a user is interested in locating some business near his/her current location. For example, the user may just say *"Find a movie theater around here"*. In such situations, the system needs to access the yellow pages server to find the list of movie theaters, interact with the user to identify the one of interest, and then query the map server for maps and/or directions. The phone number of the identified business can also be provided to the user on demand.

6. CONCLUSION

We have presented four major speech recognition applications applicable to the automobile environment ranging from continuous digit recognition to car navigation. With our adaptation techniques, we can demonstrate sufficiently accurate and robust performance for recognition in this environment. We anticipate that this superior recognition performance coupled with meaningful applications in the automobile will allow the mobile user more natural and ready access to valuable information.

7. REFERENCES

[1] C. T. Hemphill. Dagger: Real-time, software-only speech recognition. Technical report, Texas Instruments, 1993.

[2] C. T. Hemphill and Y. K. Muthusamy. Developing web-based speech applications. In *Eurospeech*, 1997.

[3] Java Website. http://java.sun.com/.

[4] Java Speech Application Programming Interface. http://java.sun.com/products/java-media/speech.

[5] Y. Gong. Source normalization training for HMM applied to noisy telephone speech recognition. In *Eurospeech*, 1997.

[6] M. T. Marx. Towards effective conversational messaging. Master's thesis, MIT, June 1995.

[7] R. Agarwal. Towards a PURE spoken dialogue system for information access. In *Proceedings of the ACL/EACL Workshop on Interactive Spoken Dialog Systems*, 1997.

[8] R. Haeb-Umbach. Robust Speech Recognition for Wireless Networks and Mobile Telephony. In *Eurospeech*, 1997.

[9] D. V. Compernolle. Speech Recognition in the Car: From Phone Dialing to Car Navigation. In *Eurospeech*, 1997.