Speech recognition in a multi-modal health care application: a worked design study

by

Jason D. Williams, AT&T Labs – Research

Annette Liska, Gomoll Research + Design, Inc.

David Suendermann, SpeechCycle, Inc. and DHBW Stuttgart

Silke Witt-Ehsani, Fluential, Inc

Table of Contents

Introduction

The healthcare field presents a host of opportunities for Voice User Interfaces (VUIs), ranging from a dictation application for surgeons during procedures dictation medical record information to an automated system which calls patients to check whether they are taking medication properly. Despite the potential, it is crucial that the strengths and weaknesses of speech technologies are sufficiently understood and appropriately matched to each application.

We view the successful marriage of technology to an application as a design task. In this whitepaper, we present a worked design study of a voice user interface for a healthcare application. This worked design study does not report on an implemented system, but rather illustrates a decision process undertaken by a VUI

designer with sight of the technical strengths and limitations of spoken language interfaces. Although any single case study cannot cover all aspects of speech technology relevant to a field, we believe this design study nonetheless provides a good general introduction to the problem of VUI design for healthcare applications.

Context and aims

The medical context of interest here is a procedure called "minimally invasive ablation", where trained clinicians and their technical assistants work in teams to destroy diseased tissue. For the purpose of this case study, a physician carefully places energy delivery devices into the patient, while the technician receives directions from the physician to drive a control panel on the machine. At present, the procedure is conducted on two large touch interfaces with different content for the physician and the technician. The displays contain critical anatomical information and data, which is best interpreted by the users when the interface is uncluttered by functional commands. For the physician, accurate decision making is dependent on real-time visual evidence and statistics; in turn, the physician then communicates those procedure decisions and next steps to the technician.

There are several problems with the status quo. First, during the procedure, the environment must remain sterile; even though the users are gloved and equipment veiled, protocols for sterile interactions necessitate as little touching between equipment and patient as possible. Second, the hands of the physician are occupied when placing devices, requiring spoken commands to the technician to set the dosage for energy delivery in the touch interface. Third, the decision making process during procedures is very fluid, and navigating to different commands, scans, and data with traditional touch interface commands can inhibit and slow down a natural workflow.

Adding a speech interface could enable dosages of energy to be set by voice, and allow anatomical scans to be generated, searched, and reviewed during the procedure. Both of these solutions would enable the physicians' and technicians' eyes and hands to remain focused on the patient's anatomy and on the position of the devices when necessary, thus delivering more efficient patient treatment.

The aim of this paper is to review some of the design considerations for creating a speech interface for this medical procedure.

Four design considerations

The remainder of the paper turns to the following design considerations:

- Audience: How do users know what to say?
- End-pointing: When is the user speaking?
- Accuracy: How can misrecognitions be mitigated?
- Accuracy: How can misrecognitions be handled when they do occur?

Audience: How do users know what to say?

Users of this system are medical technicians who use the equipment repeatedly, in a professional setting. Thus, it is possible to **train users**. In fact, in the medical field, doctors are often trained to dictate medical

record information. This use of speech recognition technology is well established at over 50% of all hospitals. This means that the targeted user group for this example would have had exposure to the general technology. The potential users are also familiar with the concept of being trained to use such a system. A common practice for users of dictation software in the medical world is to create macros that are essentially short phrases for a couple sentences for a given medical condition.

This exposure to speech technology that requires training is in contrast to a system where it is not feasible to train users, such as a telephone IVR, which may be called infrequently. In addition, this application includes a large visual display which readily allows for **available commands to be shown** on the screen or to provide priming information by showing example statements.

End-pointing: When is the user speaking?

An important task in a speech recognition interface is to determine when the user is speaking to the system. The system should only act on direct commands from the technician – it should ignore other speech, such as conversation between the technician and patient. Two solutions are possible. First, in a **push-to-talk** interface, the user presses some kind of button to indicate that they are speaking to the system. Some mobile phone applications such as Siri, Bing voice search, and Google Voice Actions use a push-to-talk interface. Second, in a **key-phrase** system, the system is constantly listening, but only reacts when it hears a key phrase. The Microsoft Xbox employs this approach. In general, a push-to-talk interface is more reliable, because a physical button press is an **unmistakable** event, whereas spotting a key phrase is itself an imperfect process with possible false alarms or false rejects.

In this system, a push-to-talk interface seems preferable. Two locations for the button are possible – first, on the equipment already in the technician's hands, or a floor pedal.

Additionally, the system needs to determine when the user is done speaking. This can be achieved via in this case **push-to-stop** or by using speech recognition algorithms to automatically detect the end of speech. Just like push-to-talk, push-to-stop is more reliable while automatically detecting the end of speech is easier from a user's point of view.

Also, in terms of usability, a well-known practice of using "earcons" to indicate when a system is processing, when a system has successfully recognized something, and so forth, has been shown to be helpful to users by helping them to be oriented as to what is happening without having to even look at the screen.

Accuracy: How can misrecognitions be mitigated?

Good speech recognition accuracy depends in part on a variety of environmental factors. First, **background noise** is important. Procedures are performed in a quiet room which yields much higher speech recognition accuracy than a noisy environment such an ambulance. Moreover, a good microphone can be used to capture **high-quality audio** – this will also contribute to better accuracy compared to telephone-quality audio which removes high frequencies and results in lower accuracy.

The positioning of the microphone is also important. The highest recognition and noise tolerance is achieved when the user is wearing a headset with the microphone positioned in front of his mouth. If the microphone has to be located on the device that has the screen for the application, then an array microphone could be used that is capable of filtering out the targeted user via his/her location against other voices in the same room. An example of a successful implementation of such an array microphone is Microsoft Xbox. As mentioned above, technicians are regular users who can be trained, maximizing the likelihood that users will be **aware what phrases can be understood** by the system. Finally, since technicians are repeat users, the speech recognition models can be **adapted to each user's voice**. All of these bode well for good speech recognition accuracy.

Lastly, recognition accuracy can be controlled by the use of a confidence threshold, that is if the system is not certain enough it understood the user correctly, it will ask the user to repeat himself rather than risking a misunderstanding.

Accuracy: How can misrecognitions be handled when they do occur ?

Despite the potential for comparatively good accuracy in this application, speech recognition is a statistical technology that will **always make some errors**, and these must be accounted for in the design. The standard way to avoid acting on a speech recognition error is to **confirm user input**. Although confirmations can substantially reduce errors, unfortunately they slow down the interaction and, hence, can be annoying to the user.

For **setting dosages and similar critical procedures**, it is crucial to avoid errors – an incorrect dosage could be dangerous to the patient. So, for such procedures, we suggest **always confirming the dose**, both visually on the screen, and with a voice prompt. Further, "yes/no" recognition can also make errors, albeit usually less than 1% of the time. However, to ensure patient safety, the reply from the technician should then be done with a button, located either on the equipment or as a floor pedal. Additionally, accuracy can be improved by implementing smart algorithms that check the validity of recognition results against past examples or history. In the case of dosage settings, that would mean only recognizing appropriate dosage amounts for the drug in question and given the available patient information.

For **reviewing past scans**, errors are much less costly. After the technician requests a scan, the search criteria can be repeated back, and the relevant **scan can be shown without confirmation**. If the wrong scan has been retrieved, the user can simply make the request again.

Conclusions and next steps

This case study has addressed the healthcare application of minimally invasive ablation, where the current interface systems are inhibited by the need for sterile, hands-free conditions. In addition, existing technology requires spoken communication between the physician and the technician to execute commands, missing an opportunity for those spoken commands to be interpreted by the device directly. Speech-enabling this interface is attractive because of the ability to free the hands and eyes from command execution, so that they are directed to the patient and to the energy delivery equipment, thus facilitating more accurate and efficient results for patient care.

This case study covered four central questions for a speech-enabled multi-modal application, dealing with user expectations, determining when users are speaking, and accuracy issues. Although the set of issues covered is not exhaustive, we believe this case study provides a good illustration of how to approach design for a speech-enabled, multi-modal application.