Designed for Enablement or Disabled by Design?
Choosing the Path to Effective Speech Application Design

Jennifer Lai[a]
Savitha Srinivasan[b]

[a]IBM T.J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532

[b]IBM Almaden Research Center, 650 Harry Road, San Jose, Ca 95120

ABSTRACT

Speech technology is often touted as the great equalizer, simplifying computer interfaces and making computers accessible to all. Since speech is a form of communication that we learn early and practice often, many have claimed that speech is by it's vary nature, a totally intuitive way to interact with computers. When computers were first introduced, the required interface was a series of arcane typed commands, known only by a small percentage of the population. With the arrival of the graphical user interface (GUI), computers became more accessible to a larger number of people since commands could be found (at worst) by an exhaustive search through the dropdowns in the menu bar. However with a spoken interface, in many ways we are back to the old model of hidden commands. If you hand somebody a phone and tell him there is a computer on the line, waiting for him to speak, the user is immediately confronted with the question of what is an acceptable command for the system. The job of helping the user past this first hurdle and onto the path towards successful task completion falls to the speech interface designer. Effective speech application design depends on the use of carefully crafted system prompts that cue the user what to say, as well as the use of a grammar designed with an understanding of how users speak in the domain of the application. Design considerations also depend on the type of speech application being created.

1. SPEECH RECOGNITION APPLICATIONS

Traditionally, speech recognition applications may be seen as belonging to one of three categories: dictation systems, navigation or transactional systems and multimedia indexing systems. Each of these application areas offers a set of benefits to the user along with a set of design challenges imposed by the limitations of the technology. Design requirements often vary significantly based on the targeted user group and usage scenario. In the first two categories, speech recognition is used as an input mechanism, occasionally in combination with one or more traditional input modalities such as a keyboard, mouse or telephone keypad. The design issues in this case revolve around maximizing the potential for accurate input and repairing errors easily when they do occur. In the third category, audio-indexing applications, speech recognition can be utilized in two ways. First, it can be used as an input mechanism to recognize the spoken query (as in the first two application categories). Secondly, speech recognition can be used to transcribe the audio contained in the multimedia recordings (e.g. broadcast news or meeting notes). The words in the transcript are used to index into the media so that spoken keywords can serve as search terms to retrieve the relevant audio/video segments. The ultimate goal in audio-indexing applications is to search through an multimedia collection by voice. However, since the speech recognition accuracy is often limited by a multitude of factors such as recording fidelity, speech spontaneity (e.g. umms and ahhs, elongation and shortening of syllables), the audio domain (for example technical versus general), and background noise conditions, the current speech recognition technology is only just beginning to make textual queries on multimedia documents practical. The design of such systems must therefore, support the limits of the technology to yield acceptable accuracy in retrieval while correctly managing user expectations.

2. SPEECH APPLICATION DESIGN

Creating a speech-only interface presents design challenges that are different from those presented in a purely graphical environment. While some basic design rules apply equally to the field of Voice User Interfaces (VUIs) as they do to GUIs (e.g. know your user), there are additional restraints imposed on the speech application designer which stem from the need to work around the limitations of the technology. Designers new to the process of creating speech applications often find themselves at a loss because there are no concrete visuals to sketch. Designing speech interfaces involves understanding conversational styles and the different ways that people use language to communicate. In this paper we will discuss the fundamental issues relating to the practice of sound design in each of these three speech application classes. We will also discuss ways of involving users along the way.

2.1 Dictation Applications

With dictation applications the words spoken by a user are transcribed verbatim into written text. Such applications are used to create text such as personal letters, business correspondence, or even email messages. Usually the user has to be very explicit, specifying all punctuation and capitalization in the dictation. Examples of commercial products for these applications are L&H Dragon Naturally Speaking, and the IBM ViaVoice product. Dictation applications are often referred to as multi-modal applications since they combine the traditional GUI modalities (visual output, mouse and keyboard input) with spoken input and often, spoken output. While many of the technological hurdles in large vocabulary speech recognition have been vanquished (continuous speech, speaker-independent, modeless products are now the norm), using speech to create text can still be a challenging experience. Many users have a hard time getting used to the process of dictating (whether it be to another human or to a machine) and the most successful ones often compose their thoughts before switching the microphone on.

As the user speaks, the text appears on the screen and is available for correction. Correction can take place either with traditional methods such as mouse and keyboard, or with speech. The problem with correcting a speech recognition error with speech, is that it is likely to be wrong the second time as well. This is especially true if the spoken word is not the system's vocabulary. Most speech systems can only "understand" or "hear" words that are contained in the vocabulary. A study that looked at error-correction patterns (Karat, 1999) for these dictation systems found that novice users can get caught is a cascade of errors when correcting an error. Part of the explanation for this is that novices were more likely to use speech to correct the error than experienced users, who tended to fall back on keyboard usage for error correction. Novices also often felt compelled to "correct work as they go" rather than dictating a portion of text and then dealing with corrections after the fact. This further broke up their ability to compose text, and occasionally confounded errors.

When designing a dictation application, especially a niche application (e.g. dictation for radiologists) the designer will do well to first observe the users in the environment where they will be dictating. There are many environmental factors that will affect how well the application will work for the particular user group. In the case of radiologists, ambient noise is a big challenge. Radiologists dictate in a reading room which has several doctors dictating at the same time, support staff moving in and out of the room, as well mechanical noise from large light boxes that display the films needing interpretation. Other user-specific issues are the doctors' willingness and motivation to switch to a speech-based application, as well as levels of computer literacy and acceptance. While a dictation application is best suited to a group of users that is already comfortable dictating (e.g. doctors or lawyers), a study of the design and usage of a dictation application for radiologists (Lai, 1997) found that not all doctors embraced the use of speech technology. Even though the application was successful in decreasing the turnaround time on reports in one hospital from 50 hours to a few minutes, some doctors could not accept the change in roles that it required. Radiologists objected to assuming the additional task of editing and correction that had previously been the responsibility of transcriptionist.

In addition to observing and interviewing users on location prior to the design of the application, field trials are recommended as soon as the dictation application is functional. Speech recognition accuracy (i.e. the number of words that are correctly recognized) can usually be improved by collecting data on location with the user group and refining the data models that drive the recognizor. Domain specific language models, and acoustic models tuned to the circumstances of usage go a long way towards assuring the highest level of accuracy possible. Another potential problem that needs to be evaluated through usability trials is how well users can detect what mode they are in. Dictation applications have two modes. In the first, everything a user says is transcribed verbatim. In the second, speech is used to tell the application what to do (e.g. "open my reports"). It is not uncommon for users to think they are in command mode when they are actually in dictation mode, and vice a versa. The chaos that results from this confusion is easy to imagine; the application jumping from one command to another as it tries to make sense of the stream of dictation.

Field trials with multi-modal applications are helpful because they allow the designer to observe how effective the visual feedback is with regarding to spotting recognition errors when they occur. While it can be cumbersome for the user to correct errors, it can sometimes be worse if the error goes undetected. Take for example a radiology report where the dictated text of "there are no signs of cancer" is instead recognized as "there are signs of cancer" (referred to as a deletion error). The two primary types of recognition problems that need to be communicated to the user are recognition failures and errors. When a recognition error occurs, the system believes that it has understood the speech, but it understands incorrectly. In the case of a recognition failure nothing happens at all. Since feedback of recognition failures is useful, designers should evaluate through user trials if the feedback mechanism they selected is working effectively. If a user asks the system to do something and the system fails to understand the command, feedback should be immediate. Both visual or auditory methods can be used, or some combination of these. While the user will eventually determine that the requested action was not taken, precious time is lost while the user sits there expectantly waiting for the system to respond correctly.

Finally, field trials for dictation applications are useful in determining if the users are having problems with the microphone. Reductions in accuracy due to such problems are very common. While great progress has been made in developing noise-canceling microphones which do a great job of dealing with average levels of background noise, these microphones are still very sensitive to problems related to the position of the microphone in relation to the mouth. If the user is holding the microphone too close to the lips, or too far, or has the microphone angled away from the mouth, there will be a sharp drop in the accuracy rate.

2.2 Navigation or Transactional Applications

Unlike dictation applications, speech is used in transactional (sometimes referred to as command & control) applications to navigate around the application or to conduct a transaction. For example, in this category of applications, speech can be used to purchase stock, reserve an airline itinerary, or transfer bank account balances. It can also be used to follow links on the web or move from application to application on one's desktop. Most often, but not exclusively, this category of speech applications involves the use of a telephone. The user speaks into a phone, the signal is interpreted by a computer (not the phone) and an appropriate response is produced.

For designers, this category of applications often presents some of the greatest challenges. Unlike desktop applications, where the interaction designer has a entire basket of tools at hand for communicating information to the user (e.g. choice of color, font, dialog boxes), with transactional speech applications, the designer is limited to the creation of the system prompts. Careful crafting of the prompts, (i.e. what the system says to the user) is critical to the success of the application. Not only are the prompts required to drive the interaction to a successful conclusion, but the prompts are also the only way to cue the user as to what can be said.

Transactional applications usually rely on the use of a grammar. A grammar is the explicit definition of the phrases and sentences that will be "understood" by the recognizor. If a user speaks outside of the grammar his sentence will be not be "heard" (i.e. the engine will return a failure) or some other sentence will be returned. One that is legal but that the user did not say. The software designer needs to make sure that he or she considers and defines all the possible ways that a person could say things. The other problem with grammars is that if the user pauses for too long while speaking, the sentence will be rejected even if a legal construct was used. Given the need to speak within the grammar, one of the hardest things in the design of transactional speech application is letting users know what they can say.

There are two basic styles of prompts. The first style is often referred to as explicit or directive. In this case, the interaction designer directs the user to say certain words or phrases. For example, "*Say next, delete, or reply*." Even when the key word "say" is not present in the prompt, explicit prompts tend to elicit a single piece of information from the user, as in the prompt "*call whom?"* or "*what city are you leaving from*?" Systems that use this style of prompts are called directed-dialog systems. Most systems that have been built to date use explicit prompts since this is the easiest way to deal with unpredictable accuracy levels. This type of prompt helps to constrain what the user says and is appropriate to use when either the cost of making an error is high or the recognizor can only handle a small vocabulary robustly. Explicit prompts however, do not make for the most natural type of interaction. On the other end of the spectrum, are the conversational speech systems which use implicit prompts. These prompts provide a more natural interaction. A conversational airline reservation system might start off by asking the user what his travel plans are. The tradeoff, of course, is accuracy. A conversational prompt is more open-ended and therefore the likelihood of an error occurring is higher. For example starting an application with the prompt: "*How can I help you*?" would most likely generate a high degree of errors. This invites people to say just about anything. A more constrained prompt might be "*Which flight are you interested in checking on?* " The system must then be ready to accept phrases such as "*I'm interested in a morning flight from New York to Los Angeles."* In addition to these two primary categories of prompts there are other important prompting techniques such as tapering prompts or incremental prompts (Yankelovich, 96). In the first technique the designer shortens the prompts over time, assuming that the user needs less information as he becomes more familiar with the interaction, and in the second, additional information is provided as the system deems that it is necessary.

The best way for a designer to determine if she has done a complete job defining the grammar is to spend time listening to the users speak in the domain of the application. Often the application that is being designed already exists in some other form. Perhaps a voice response unit is currently being used, or a portion of a call center application is being automated. Listening in on the calls helps to define the tone of the interaction as well as the vocabulary that is used. Once a trial grammar is in place, the designer should have users test it to see which common constructs have been forgotten. Sometimes, even before the grammar and the speech system are implemented, trials can take place using a Wizard of Oz (WOZ) system. With a WOZ, the users are led to believe (or asked to pretend) that they are interacting with a fully functioning speech system. In reality, the "wizard" is listening to the speech and generating the appropriate system response. When using a WOZ it is important to remember to simulate several speech recognition errors. Since graceful error recovery is an important part of every speech system, this should be a well exercised path in the design !

2.3 Multimedia Indexing Applications

In multimedia indexing applications, speech is used to transcribe words verbatim from an audio file into text. The audio may be part of a video. Subsequently, information retrieval techniques are applied to the transcript to create an index with time offsets into the audio. This enables a user to search a collection of audio/video documents using text keywords. Digital audio and video are becoming increasingly popular such that large collections of multimedia documents can be found in diverse application domains such as the broadcast industry, education, medical imaging, and geographic information systems. Retrieval of unstructured multimedia documents is a challenge today and requires content-based retrieval where the content of the document is examined for the presence or the absence of an object, of words or phrases, or a visual action. Therefore, cataloging and indexing of audio/video has been universally accepted (Wactler, 1999) as a step in the right direction towards enabling intelligent navigation, search, browsing and viewing of digital audio/video.

One approach to multimedia retrieval is to apply image retrieval techniques to key frames extracted from the video. In this approach, an image is posted as a query, and similar images are retrieved. There are two reasons why this approach, in general, has not become popular yet . First, in most practical situations the user does not have such an image handy to formulate the query. Second, the state of the art in content-based image retrieval has not yet reached the semantic level desired by most users. Rather, it is typically done in a feature space, such as color histograms, color layout, color blobs , texture and shapes (Flickner 1995). A more popular approach to video retrieval is to search the audio transcript of the video using the familiar metaphor of free text search (Jones, 1996; Srinivasan, 2000). The indexed transcript provides direct access to the semantic information in the video.

While searching the audio using text keywords proves to be quite efficient, browsing the video is much more time consuming than browsing of text. This is because the user has to play and listen to each of the retrieved videos, one by one, unlike with text where a quick glance at the result page is often sufficient to filter the information. In this case, it is more efficient to browse the visual portion using a video segmentation technique (e.g. a video storyboard). A few pages of storyboard, each showing ten or more key-frames, can cover one hour of video by showing the main visual scenes contained. Therefore, a popular approach to multimedia retrieval is *"Search the speech, browse the video"* where the video and audio are treated as two parallel media streams of information that are related by a common time line. Thus, the audio stream is used for searching and the video stream for quick visual browsing in a complimentary manner to provide the desired video search functionality.

A well known issue in speech indexing is the concept of in-vocabulary terms and out-of-vocabulary terms. This corresponds to the words that can be "understood" or "heard" by speech applications. In-vocabulary terms can be understood by speech systems, and therefore they can be retrieved using keyword search. In contrast, out-of-vocabulary words cannot be "understood" and will be misrecognized instead as a similar sounding in-vocabulary word. This implies that out-of-vocabulary words cannot be retrieved using text keywords. This problem is typically addressed by creating an index of the sub-word or phonetic representations of a word. When presented with a text keyword, it is first translated into its equivalent phonetic representation and the corresponding phonetic index is searched. The accuracy of phoneme recognition however, is limited, particularly in the case of short words (Ng, 1998). Therefore, in practice, combined indexes that comprise of a keyword index and a phonetic index are used to provide the best search performance.

It has been shown that word error rates can vary between 8-15% and 70-85% depending on the domain and tuning of the recognition engine. The 8-15% error rates correspond to standard speech evaluation data and the 70-85% corresponds to "real-world" data such as a one hour documentary and commercials. In general, it has been shown that for an speech recognition error rate of about 30%, a retrieval system can achieve about 80% of the effectiveness of text search engines that operate on perfect text documents (Wactlar, 1999). This has been validated for multimedia collections of a few hundred hours and is as yet unknown for larger document collections.

It is important to have realistic expectations with respect to retrieval performance when speech recognition is used. The user interface design is typically guided by the *"Search the speech, browse the video"* metaphor where the primary search interface is through textual keywords, and browsing of the video is through video segmentation techniques. While the requirements of specific applications vary, our experience indicates that the precision of search results are more important to a user than the recall, i.e. the accuracy of the top-ranked search results is more important than finding every relevant match in the audio. Therefore, the ranking of search results may be biased to address this. In general, since the user does not directly interact with the indexing system using speech input, standard search engine user interfaces are seamlessly applicable to speech indexing interfaces. However, the following design guidelines can result in a more successful speech indexing system: First, since the transcript of the speech is not accurate enough to result in fully readable grammatical text, it is not advisable to display the entire transcript as part of the search results since it can cause a negative impression on the user. Secondly, search interfaces can "guide" the user in the selection of search terms by providing a list of search terms from the transcript.

3. CONCLUSION

Successful speech applications, like other types of applications, need to incorporate both an understanding of the user and the circumstances of use into the design. Each of the categories discussed in this paper emphasizes the need to pay attention to the different characteristics of the interaction when doing system design. The primary difference between speech applications and other applications which use a more mature technology (e.g. keyboard input) is that the designer must take into account the limitations of the technology. Without an understanding of what causes errors in speech, and a focus on both error prevention strategies and graceful error recovery, the resulting application will be unusable at best. For dictation applications, several strategies apply. First of all, study the users in the location where they will be dictating. Pay particular attention to ambient noise levels, and frequency of interruptions. Design the interface to support these aspects of the interaction, as well as supporting the users task requirements (e.g. eyes-free operation). For transactional applications, the designer must understand and reconstruct through the grammar the way the users speak in the domain. Observations, interviews and a wizard of oz setup are all helpful in achieving this goal. Finally, for multimedia indexing applications, the most important design consideration is optimize speech search performance combined with synergistic user interface considerations that further maximize this.

References

1. Flickner, M. et al. (1995), Query by image and video content: The QBIC system, IEEE Computer, Vol 28, No. 5, pp.23-32.

2. Jones, G. J. F., Foote, J. T., Spärck Jones, K., and Young, S. J. (1996), Retrieving Spoken Documents by Combining Multiple Index Sources. In *Proceedings of SIGIR 96, pp. 30-38*, Zurich, Switzerland.

3. Karat, C., Halverson, C., Horn, D., and Karat, J. (1999), Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems. In *Proceedings of CHI '99: Human Factors in Computing Systems*, pp. 568-575, Pittsburgh, PA.

4. Lai, J., Vergo, J. (1997), MedSpeak: Report Creation with Continuous Speech Recognition. In *Proceedings of CHI '97: Human Factors in Computing Systems*, Atlanta, GA.

5. Ng, K. and Zue, V. (1998), Phonetic Recognition for Spoken Document Retrieval. In *Proceedings of ICASSP 98*, pp. 325-328.

6. Srinivasan, S. and Petkovic, D. (2000), Phonetic Confusion Matrix Based Spoken Document Retrieval. In Proceedings of SIGIR-2000, Athens, Greece.

7. Wactlar, H., Christel, M., Gong, Y. and Hauptmann, A. (1999), Lessons Learned from Building a Terabyte Digital Video Library. In IEEE Computer.

8. Yankelovich, N. (1996), How do Users Know What to Say? In *ACM Interactions*, Volume 3, Number 6, November/December.