UNDERSTANDING SPEECH WITHOUT RECOGNIZING WORDS

Christopher Schmandt
Speech Research Group, Media Laboratory
Massachusetts Institute of Technology

Abstract

This paper describes a system to exploit non-lexical acoustic cues to listener comprehension in a dialog between a human and a computer. The computer uses text-to-speech synthesis to recite a series of driving directions. It classifies the listener's responses as affirmative or negative based on duration, pitch, and energy; this is used to control flow of the conversation to facilitate the listener's task.

1 Speech as a Control Channel

Speech is used for communication, in a process whereby a talker wishes to cause a listener to do something or change state in response. The talker may intend that the listener perform some action for example, or verify that some action has been performed, or supply or receive some information. If supplying information, the talker's motivation is not simply to recite data, but rather to confirm that such data was received and understood by the listener. Without confirmation it is not possible to satisfy the talker's intention that the listener know the information.

Speech may be used in this context as a data channel or as a control channel, usually interleaved. It is the role of voice as a control function which is used to guarantee receipt of the message. This function may be accomplished by combinations of explicit words ("O.K.", "What did you say?"), by pauses, either empty or filled ("ummm...") or other paraverbals ("huh?", "uh-huh!"), and prosodics or intonation (pitch contour, syllable duration, and relative energy levels). In face to face interaction, one may also use facial or body gestures and eye contact to indicate one's state of understanding or attention, but this paper is concerned with voice only situations, such as use of a telephone.

Proceedings, American Voice Input/Output Society Conference, AVIOS, 1987

2 Acoustic Cues to Comprehension

One of the common uses of voice as a computer interface is to supply some information to a listener at a remote location, usually by telephone. Speech synthesis or digital audio playback may be used to present electronic mail, voice messages, inventory order status, traffic, weather or financial information, among a wealth of applications. For each of these, there is some function needed to specify exactly what information is required, usually by touch-tone input. What is often missing is any control function to confirm that the message was actually understood.

If we consider this transaction as part of a communication act, we would expect the talker to obtain some confirmation from the listener that the message was understood. Such interfaces are currently fairly awkward. The most obvious is to require a touch tone response to each paragraph, or chunk of new information. This explicit confirmation slows down the interaction and is certainly not intuitive.

Another alternative would be to use conventional speech recognition devices. These suffer from a number of drawbacks, however: they usually do not work well over the telephone, they are often speaker dependent, and they have a limited vocabulary. The latter implies that even if there were no acoustic problems with speaker independent recognition over the telephone network, one would still have to instruct a user in a set of *rules*, or legal words to use and when to say them. We would prefer to instead discover the discourse rules humans use and then try to build computer systems to emulate them so the interaction is natural and intuitive.

The technical limitations of speech recognition force us to seek other acoustic approaches to understanding the listener's responses rather than trying to recognize a few words. Observation of recorded dialogs between people further suggests that word recognition may have limited utility because of the variety of paraverbal and intonational responses which are used, in addition to a large number of words (which fail simple categorization such as yes/no). Lexically, the control function may be much less explicit than other speech functions.

3 Dialog Structure in Direction Giving

The domain we have chosen to work with is that of the computer giving driving directions. We chose this domain in part because of previous work [1] and in part because of its computationally tractable discourse structure. In this scenario, the computer uses text-to-speech synthesis to give directions to a caller, who is presumably writing them down.

Our observations of humans giving directions suggests that talkers break the directions down into logical segments, or paragraphs, each containing a relatively simple set of instructions between significant landmarks. The talker pauses between each segment, which allows the listener a chance to respond.

We group the responses into four classes:

- *none.* The listener says nothing. After a suitable timeout period, (a function of the complexity of the most recent outgoing data utterance), the talker assumes understanding and continues with the next paragraph. After several successive silences, the talker will probably engage in *channel checking* behavior [2], asking, for example "Hello, are you there? Can you hear me?".
- *affirmative.* The listener indicates understanding. Examples would be "O.K", "uh-huh", "yup", "Yes, I know where that intersection is..." or "...right after the third light. O.K."
- *negative.* The listener indicates lack of understanding explicitly. Examples: "Take a right *where?*", "But I thought I was going to Cambridge.", "I'm totally lost now!", "How will I recognize Kendall Square?"
- *timing.* The listener indicates a timing problem, usually needing more time to write down the directions. Examples: "Could you hold on a moment?", "Just a second", "Repeat that last part please."

For our purposes, the first two are both treated as affirmative responses, although if the response is silence we note it and engage in channel checking after three successive silences.

The latter two can both be treated as negative responses, on the assumption that it is always better to repeat known information than to skip possibly

confused portions. We don't believe we can distinguish negative and timing responses, but both are dealt with by triggering a repetition, perhaps with more detail, of the previous paragraph. If the caller simply needs more time to write down the paragraph, repetition does not interfere.

4 Duration as a classifier

The problem is to differentiate these two or three classes of events acoustically. The first pass, implemented at the time of this writing, is based on utterance *duration* only. We believe that many of the affirmative responses will tend to be quite short, and the negative or timing responses will be longer in length.

An average magnitude function applied over a 100 millisecond window is used to detect energy exceeding a background noise level. Once this level is exceeded, the utterance is timed until the energy falls below the threshold for another period of time. Then the utterance is judged for length; less than about 600 milliseconds is "short" or affirmative.

The audio processing is done on an audio server processor (the "grunt detector") which communicates to its host (a Sun 2) over a serial line. The protocol adopted so far allows calibration, magnitude calculation and duration computation, and synchronization. The latter is necessary for the host to indicate when the server should listen; because of the telephone line interface, outgoing audio is also detected on the incoming audio line. This makes interruption (by the human) currently impossible, a serious shortcoming.

5 Direction Giving

For our development environment, we assume a single set of directions, i.e., how to get to a local bakery from the M.I.T. campus. The computer greets a caller and asks a few questions; from our prior work [3] we know that callers are very likely to answer whatever question they are asked. This serves two purposes. First, it enables us to calibrate the audio levels to determine the caller's amplitude and the line's background noise level.

Second, it gets the caller used to talking back to the system.

Next, the directions are given as a series of paragraphs, and the audio server is asked to monitor after each for a possible reply. Short (affirmative) replies cause immediate recitation of the next paragraph. Long (negative) replies cause repetition of the previous paragraph. The first repetition consists of the same text played at a slower rate. The second negative response causes selection of a more descriptive version of the same information.

In the case of no reply after a suitable timeout, the next paragraph of directions will be generated. After three successive timeouts on silence, the programs asks "Are you still there? Can you hear me?" and hangs up if there is no response.

6 Additional Acoustic Classifiers

The work described so far has been completed to date. What follows is currently in progress, and will probably be completed by the time the paper is presented.

Although the basic duration classifier works for a surprisingly large number of utterances from a range of speakers, there are several pivotal discourse events in which it breaks down. These are short questions, e.g. "What?", "Where?", and echo sentences which probably mostly serve as timing place holders, e.g. the *listener* repeating "...take a right after the Longfellow Bridge...".

The first of these is the most important, in terms of our desire to maximize the likelihood that the listener will receive the message, perhaps at the price of needless repetition. Note that we are trying to detect *short* questions; long ones will already trigger repetition. We observe that questions have rising terminal pitch, and the further constraint of duration indicates that short questions must have pitch rising throughout as there simply is not enough time for the intonational gesture otherwise. Thus, we hope to detect short questions using real time pitch tracking to find this monotonically increasing pitch contour.

The echo responses may be detected by energy level. We observe that echoing is usually done at a lower magnitude than a question or request

for more information. We calibrate energy levels to a particular speaker by asking a question requiring a neutral declarative response early in the conversation, and hope to be able to then define a lower level below which we assume an echo.

Note that the latter case is an example where too little information may be presented by mistake. We should note when this happens, and back up gracefully. For example, if the response to utterance N seems to be an echo, but the response to utterance N+1 is definitely a rejection, it is probably best to play both paragraph N and N+1 together in response. This will further help synchronize talker and listener.

7 Interruption

Interruption over a telephone line is always a difficult problem, mostly due to the two wire nature of the transmission medium. Whatever speech we put out on the phone line comes back as well, hopefully somewhat attenuated. This always makes it difficult for a computer speech system to allow interruption, unless it uses touch-tones, which can be detected clearly over outgoing speech.

Interruption may be more manageable when it is to be classified only as a single event, rather than attempting to recognize particular words in the interruption. For our purposes, it is adequate to detect that an interruption has occurred. The appropriate behavior is to pause immediately, then probably to back track and repeat the previous plus current paragraph. If the interruption occurs late in the current paragraph, however, we may be able to correlate it more closely to specific pieces of information being transmitted by the speech synthesizer.

We hope to be able to detect interruption as an acoustic event against a variable background noise level (the outgoing speech), perhaps using some echo cancellation hardware. Note that we are not trying to *recognize* the interruption audio, just to *detect* its occurrence.

8 Acknowledgments

Thanks to Lorne Berman for writing the main logic of the direction giving software, to Kevin Landel for work with observing human direction givers, to Mike McKenna for writing pitch and energy analysis tools, and to Jim Davis for discovering the direction giving domain in the first place.

This work was supported in part by DARPA, Space and Naval Warfare Systems Command, under contract number N00039-89-C-0406 and by NTT, the Nippon Telegraph and Telephone Public Corporation. Hardware support was provided by Sun Microsystems, Speech Plus, and Digital Equipment Corporation.

References

[1] James R. Davis and Thomas F Trobaugh. *Direction Assistance.* Technical Report, MIT Media Technology Lab, (in preparation).

[2] P.J. Hayes and R. Reddy. Steps towards graceful interaction in spoken and written man-machine communication. *Int J. Man-Machine Studies*, 19:231::284, 1983.

[3] C. Schmandt and B. Arons. A conversational telephone messaging system. *IEEE Trans. on Consumer Electr.*, CE-30(3):xxi–xxiv, 1984.