INTERACTION AND FEEDBACK IN A SPOKEN LANGUAGE SYSTEM

Susan E. Brennan
Psychology Department
State University of New York
Stony Brook, NY 11794
brennan@psych.stanford.edu

Eric A. Hulteen
Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
hulteen@applelink.apple.com

Abstract submitted to the AAAI-93 Fall Symposium on Human-Computer Collaboration: Reconciling theory, synthesizing practice

KEYWORDS: *Speech, dialog, feedback, conversational repair, agents, telephony.*

TOWARD A MORE ROBUST SPEECH INTERFACE

Traditional approaches to improving the performance of spoken language systems have focused on improving the accuracy of the underlying speech recognition technology. The assumption is that if a system can translate exactly what the user said into text and then map this onto an application command, speech will be a successful input technique. With this approach, improving speech recognition accuracy requires asymptotic effort that ultimately is reflected in the cost of the technology.

We argue that perfect performance by a speech recognizer is simply not possible, nor should it be the goal. There are limiting factors that are difficult or impossible to control, such as noise in the environment. Moreover, many words and phrases in English are homophones of other words and phrases, so in some situations, both human and machine listeners find them ambiguous. People frequently have trouble sticking to the grammar and vocabulary that a spoken language system expects. Finally, because people have many other demands on them while they are speaking such as planning what to say next and monitoring their listeners and the environment, they frequently do not produce the kind of fluent speech that a recognizer has been trained to process. Even though human utterances contain "speech errors," most human listeners hardly notice. The intrinsic limits on the well-formedness of utterances and on the accuracy of speech recognition technology suggest that to solve the problem, we must first redefine it. Let us start by considering how people handle these problems in conversation.

THE COLLABORATIVE VIEW

Having a conversation is not simply the encoding and decoding of well-formed messages. Conversation is collaborative. People coordinate their individual knowledge states by systematically seeking and providing evidence about what has been said and understood; this is the grounding process (Clark & Wilkes-Gibbs, 1986; Clark & Schaefer, 1987; 1989; Clark & Brennan, 1991). When there is a misunderstanding, both conversational partners are responsible for repairing it, and they do so by trying to expend the least collaborative effort (Clark & Wilkes-Gibbs, 1986). That is, one partner often puts in extra effort in order to minimize the effort both partners expend collectively.

In conversation, shared meanings are accrued incrementally, along with evidence of what has been understood so far. Recognizing and recovering from errors is a matter of providing the right kind of evidence to a conversational partner at the right moment. Certainly it's important to provide evidence to a conversational partner when one recognizes that there has been a misunderstanding - we will call this *negative evidence*. But evidence of what has been understood - *positive evidence* - is necessary as well. A dialog cannot proceed without it (Brennan, 1991; Clark & Brennan, 1991). Positive evidence is precisely timed and it includes short utterances with no obvious propositional content of their own, such as "uh huh" or "hm" (see Yngve, 1970 on backchannels), explicit acceptances such as "ok," and clarification questions, as well as relevant next turns, and even continued eye contact. Positive evidence is necessary because in a dialog neither partner is omniscient. At any moment, a speaker and a listener may have different takes on what has just been said. Because part of the work of repairing a misunderstanding is in identifying that one has occurred, a misunderstanding or error cannot be reliably identified unless the partners continually seek and provide evidence of understanding (Brennan, 1990).

An utterance by itself is not a contribution to a conversation, for it is always possible that the addressee didn't know she was being addressed, or that she didn't hear the utterance, attend to it, or understand it. Contributions to conversation are constructed by speakers and addressees acting together. In modeling the process of jointly contributing to dialog, we follow Clark and Schaefer's (1987, 1989) contribution model. Each contribution to a conversation consists of two phases: a presentation phase and an acceptance phase. Every utterance is a

presentation. For a speaker to conclude that her utterance contributes to a dialog, it must be followed by an acceptance phase, that is, by evidence that the addressee has understood well enough for current purposes. Likewise, when the situation involves delegating an action to an agent, the speaker needs evidence that the action was successful. Contributions have been used elsewhere as a basis for modeling the interactive structure of human-computer dialog (Brennan & Cahn, 1993; Payne, 1990).

A speaker evaluates the available evidence to determine what state a listener is in with respect to what was just said. The listener also develops a belief that she is in one of these states. In Clark and Schaefer's model (Clark & Schaefer, 1987), the possible states form an ordered list for a speaker, A, and an addressee, B, with respect to an utterance, u:

```
State 0: B didn't notice that A uttered any u
State 1: B noticed that A uttered some u
         (but wasn't in state 2)
State 2: B correctly heard u (but wasn't in
         state 3)
State 3: B understood what A meant by u
```

When two partners interact, neither has direct access to the other's beliefs or mental state. Upon hearing an utterance, listeners go about providing evidence about their state of understanding to the speaker; upon making an utterance, the speaker seeks evidence from listeners that they have understood. A problem arises when a partner believes she has failed to reach one of these states, or when she believes she has reached a state but the evidence she provides leads her partner to believe that she hasn't. Listeners display evidence that marks the state they believe themselves to be in, with respect to the most recent utterance. A listener may recognize there is a problem and initiate a repair herself. Or else the evidence she provides may not be what the speaker was expecting, and the speaker may be the one who recognizes that there is a problem. Of course, at times neither partner may recognize a problem right away, and that may lead to more problems requiring more extensive repairs later.

GROUNDING WITH A SPOKEN LANGUAGE SYSTEM

Some spoken language systems provide feedback to users, but in an inflexible and rather ad hoc way. For instance, a system can echo a user's input or ask for confirmation of every command. But users find this cumbersome if it happens all the time. In addition, informative error messages can be produced when certain kinds of processing errors occur. But this is no guarantee that, in the absence of an error message, the user won't be left wondering whether she has been understood. Our goal is to address these problems in a more adaptive, context-sensitive, and systematic way. To do this, we have extended Clark and Schaefer's (1987) proposal. The result is an adaptive feedback model for spoken language systems, based on the need for grounding in communication with a system to whom a user can delegate actions. Here is an ordered list of system states, described from the perspective of the user:

State 0: NOT ATTENDING. The system isn't listening or doesn't notice that the user has said anything.

State 1: ATTENDING. The system has noticed that the user has said something, but it hasn't interpreted the words.

State 2: HEARING. The system was able to identify some of the user's words, but hasn't parsed the entire utterance

State 3: PARSING. The system received what seems to be a well- formed utterance, but hasn't mapped it onto any plausible interpretation.

State 4: INTERPRETING. The system reached an interpretation, but hasn't mapped the utterance onto a application command.

State 5: INTENDING. The system has mapped the user's input onto a command in its application domain.

State 6: ACTING. The system attempts to carry out the command. It is not known yet whether the attempt will have the desired outcome.

State 7: REPORTING. The system may or may not have been able to to carry out the user's command, and reports any evidence available from the application domain.

States 0-2 are based on Clark and Schaefer's states 0-2, 3-4 correspond to their state 3, and 5-7 are necessary extensions for a dialog involving spoken delegation to a computer agent. For present purposes, we adopt the assumption that each state with a higher number depends on the states numbered below it. There are of course exceptions to this; for instance, if an agent were to discover that a goal it intends to carry out has already been achieved, or if an agent were able to guess or predict a user's intention without having correctly heard a command, then the agent need not pass through these states in sequence. Evidence of success at a particular level obviates the need for evidence at lower levels.

PERFORMANCE OF THE MODEL

Feedback Messages

Consider this simple exchange between a user and her telephone agent:

```
User:   "Call Lisa."
System: "I'm calling Lisa."
```

With its response, the system agent provides evidence of having passed through the states of attending to, hear-

ing, parsing, interpreting, intending to act, and acting on the user's utterance. Here are example messages associated with individual states:[1]

```
User:   "Call Lisa."
System: (one of these)
        State 1: "I'm still listening."
        State 2: "I heard something."
        State 3: "I heard you say CALL LISA."
        State 4: "I understood you to mean
                  call Lisa."
        State 5: "Do you want me to call Lisa?
        State 6: "I'm calling Lisa."
        State 7: "ok, here's Lisa."
```

Note that for human conversational partners, some of these states may be treated as equivalent due to conversational implicature (e.g., a message like that in State 5 may be used in States 3 and 4 as well); however, this hierarchy is intended to reflect the levels of processing problems that can occur in a spoken language system. Of course, an agent should not provide feedback for all these states every time the user speaks. That would be extremely tedious. In our model, a dialog partner provides feedback at the highest state of processing that it reaches.

Consider this exchange:

```
User:   "Call Lewis."
System: "I'm calling Lisa."
```

Even with experienced users and the best speech recognizer, the agent may misrecognize all or part an utterance. For a telephony application, it is critical that the user be allowed to interrupt the agent at this point, before the call goes through. This would require the user to react immediately. So when this kind of problem is likely to arise, such as when conditions are noisy or when a user is inexperienced with a recognizer's vocabulary or grammar, our model adjusts the feedback it provides:

```
User:   "Call Lewis."
System: "I heard you say call Lisa.
         I'm calling..."
User:   "No!  I said, call Lewis."
```

Although the agent appears to have reached State 6, it provides critical feedback at State 3. Informative feedback provided at an earlier state gives the user more time to interrupt the system and initiate a repair. This can be done in a variety of ways, depending on the application. The user may be able to interrupt by voice, in the case of speech recognizers that can keep listening for input while the system provides text-to-speech output, or else the user could interrupt with a key from a telephone keypad designated for this purpose. In any event, the feedback associated with State 3 has turned out to be particularly crucial for our telephony application.

Consider the situation where there are several phone numbers associated with one individual:

```
User:   "Call Chris."
System: "I'm calling Chris DiGiano at home."
```

Here the agent provides evidence of just how it has interpreted the command. Perhaps it has chosen the number where the callee is most likely to be, or perhaps it has just taken the first number from a list of possible numbers. In any event, if the user has relevant knowledge, she can interrupt at this point. Or, if the agent has no heuristic for choosing between several possibilities, it can ask for more information from the user in order to proceed from State 3 to State 4:

```
User:   "Call Eric."
System: "Do you want me to call Eric Hulteen
         or Eric Stubbs?"
```

The examples so far have shown positive evidence, where the agent finds itself proceeding successfully through the grounding states. When the agent's processing breaks down, this is marked with appropriate negative evidence as well. Here is an ordered list of error messages associated with potential breakdowns the agent can recognize and convey to the user:

```
User:   "Call Susan."
System: (one of these)
        State 0: "Still waiting for input."
        State 1: "What was that again?"
        State 2: "Call WHO?"
        State 3: "I'm sorry,
                  Susan isn't in my directory."
        State 4: "I'm sorry,
                  I don't know how to do that.
        State 7: "I'm sorry, the line's busy.
                  Shall I keep trying?
```

Error messages like the one in State 2 were used in an early spoken language interface called "Put That There," developed at M.I.T. (Schmandt & Hulteen, 1982). Without much trouble, users were able to help the system repair utterances that it had partially recognized. With the current feedback model, we have found that conceptualizing understanding by both the user and the agent as a succession of successive states provides a systematic way to program adaptive feedback. So far, we have constructed two prototypes of this model and have observed people using them. One is a simulated speech interface for Wizard of Oz studies, and the other is a prototype telephony agent application.

Prototypes

We constructed two prototypes through a sequence of informal user studies, consistent with the philosophy expressed as "user centered design" (Norman & Draper,

[1] Note that we are not committed to the wording of any of these error messages. They can be worded to be more telegraphic and less anthropomorphic. For instance, "still listening" may be preferable to "I'm still listening." See Brennan and Ohaeri (1993) for an experimental comparison of message styles in language interfaces. Message design for a particular kind of situation should depend on the application and its population of users.

1986). Our goal was to get as much information about usability as possible for the least amount of time and effort, and to immediately feed back changes inspired by users to the prototypes themselves.

Wizard-of-Oz study. First, we developed a HyperCard-based simulation of the telephone application and its interface, to support a Wizard-of-Oz study where we simulated the telephone agent with its spoken language interface. The simulation was connected to a speech synthesizer and a text parser. It simulated a two-line telephone system and provided the experimenter with a control panel for generating telephone sound effects and text-to-speech feedback. Simple functions were supported: dialing by name, dialing by number, holding, switching lines, picking up calls, and hanging up. Users were 12 employees of Apple Computer, Inc. The users role-played several telephone-based tasks. They were instructed to imagine calling their "phone agent," e.g. the voice interface to their personal office telephone, from a pay phone. They were instructed to press the # key when they wanted to address or interrupt the voice interface, and to speak into the telephone handset. There was no instruction about the kind of language the agent would understand. The tasks included: calling a coworker, placing that call on hold and calling another coworker, terminating that call and returning to the first call, answering an incoming call, and placing a call to someone at a car phone.

As subjects spoke, the experimenter rapidly typed their input into the text parser and used the buttons on the control panel to provide appropriate telephone sounds and synthesized speech messages in response. The subjects' input and the system's responses were logged for later analysis. After the session, subjects filled out a questionaire and discussed their reactions with the experimenter.

The tasks were planned so that most of the time the user's utterances would be "understood" by the system. However, several communication problems were scripted into the task, in order to see if users could use conversational strategies to get back on track. These problems were: missing one word of the user's command, completely misunderstanding the command (that is, hearing a different command), and being unable to complete a phone call because of a busy signal.

Among the results were that users did not always remember to hit the # key to address or interrupt the phone agent. They sometimes tried to interrupt verbally while the system was producing a text-to-speech message. Ideally, a spoken language system should respond to verbal interruptions while it is processing input or producing text-to-speech output. In addition, when the simulated speech recognizer consistently interpreted users' input as they intended it, some users reported that they found messages from State 3 tedious. On the other hand, after errors of mishearing, users found messages at this level helpful and depended on this evidence that their input had been interpreted as they intended. Upon hearing the system's incorrect interpretation, most were able to interrupt the system in time to make a repair. This supports the prediction from the model that feedback should be provided concerning the highest possible state the system reaches, except when it is error-prone at an earlier state.

Working prototype. We implemented a second prototype in LISP. We used a custom-built state-machine to control a telephone agent application, speech recognizer, text-to-speech synthesizer, and dialog manager. This prototype used a speaker independent, connected speech recognizer developed at Apple Computer, Inc. (see Hulteen, 1991). With a limited grammar, this speech recognizer was remarkably fast and successful at recognizing the utterances of experienced users. But it performed considerably less well with users who had no idea of the vocabulary or the grammar that had been defined for it. As we had expected after the Wizard-of-Oz study, users sometimes had problems determining when the system was waiting for input. If the speech recognizer cannot listen all the time for input, there needs to be some continuous cue – perhaps a distinct kind of audible tone – whenever it is listening. This brings up the additional point that evidence of the state of a spoken language system need not always be in the form of verbal messages; we are currently experimenting with other forms of feedback to be used in conjunction with verbal messages.

SOLVING THE GROUNDING PROBLEM IN HUMAN-COMPUTER INTERACTION

Because neither partner in a dialog can read the other's mind, communication problems tend to arise asymmetrically. That is, one partner realizes that there is a problem before the other one does. In order to initiate a repair, a user needs to take different actions depending on which state the system agent appears to have reached.

In the approach described here, the adaptive feedback provided by the agent after a user's utterance marks the state that the agent recognizes itself to be in with respect to that utterance. The idea is that if the agent continually provides feedback about the state it recognizes itself to be in, the user can construe whether the agent has understood her utterance as she intended it, and she can assess whether they both recognize the agent to be in the same state with respect to that utterance.

This feedback model is adaptive in several ways. Different feedback is given depending on the agent's state with respect to an utterance of the user's. In addition, the system keeps a history of the user's utterances and its responses. Overt error messages and attempts to initiate repairs are included in this history. Additional feedback about earlier states is provided, particularly when reaching these states has been problematic previously. For instance, if the dialog so far contains evidence of hearing errors (likely when the environment is noisy) or if parsing errors are expected (likely when the user departs from the grammar the speech recognizer has been trained on), then the agent echoes the user's input with

evidence about State 3, even though it has reached a higher state. In addition, whenever there are several possible task objects that could have been indexed by a user's referring expression, then the one chosen by the agent can be explicitly specified in its response (see also Brennan, 1988). Then the user can take appropriate action if what the agent got is not what she meant.

Task-level information can be integrated into the feedback as well. If a command is a potentially destructive one, then the agent can state its interpretation and ask the user for confirmation before acting. This would correspond to setting a higher grounding criterion (requiring stronger evidence from a conversational partner before concluding the acceptance phase to a contribution is complete). Another way to use task-level information with this hierarchical model is to keep track of completed task actions and their outcomes. When a user's goals are known, task actions often have more and less desirable outcomes. So when the agent fails to achieve the preferred outcome of a user's command, the agent can save this information and offer to try again later.

CONCLUSION

The adaptive feedback model provides a theoretically motivated basis for generating messages from agents. These messages include not only informative error messages, but also positive evidence of understanding. This approach makes it relatively straightforward to specify programmatically the kinds of feedback messages that should be provided to users and to specify exactly when to provide them. We have incorporated this model into prototype spoken language systems and are observing users interacting with them.

A system should provide the right kind of evidence at the right time, including positive evidence when things are going smoothly. If the user can depend on reliable feedback from the system, she will be better able to recognize when a problem has occurred. When the user knows exactly where the problem is, selecting a repair strategy is relatively natural.

Finally, spoken language systems can be improved by focusing on issues of dialog and feedback, rather than focusing exclusively on improvements to the accuracy of the underlying speech recognition technology. A more robust speech interface provides feedback adaptively and makes use of the human conversational strategies that users bring with them to human-computer interaction.

ACKNOWLEDGEMENTS

We thank Lewis Knapp, Lisa Stifelman, and Chris DiGiano for their creative input and participation in this work. Thanks also to Rick Parfitt and Matt Pallakoff for technical assistance, and to the members of Apple Computer's Human Interface Group and Speech and Language Technologies Group for their technical and intellectual support.

REFERENCES

Brennan, S. E. (1988). The multimedia articulation of answers in a natural language database query system. In Proc., Second Conference on Applied Natural Language Processing, pages 1-8. Association of Computational Linguistics, Austin, TX.

Brennan, S. E. (1990). Seeking and providing evidence for mutual understanding. Unpublished doctoral dissertation, Stanford University, Stanford, CA.

Brennan, S. E. (1991). Conversation with and through computers. User Modeling and User-Adapted Interaction, 1:67-86.

Brennan, S. E. and Cahn, J. (1993). Modeling the progression of mutual understanding in dialog. Manuscript.

Brennan, S. E. and Ohaeri, J. O. (1993). The effects of message style on people's mental models of computers. Manuscript in submission.

Clark, H H. and Brennan, S. E. (1991). Grounding in communication. In J. Levine L.B. Resnick and S.D. Teasley (Eds.), Perspectives on Socially Shared Cognition. APA, Reading, MA.

Clark, H. H., and Schaefer, E. F. (1987). Collaborating on contributions to conversations. Language and Cognitive Processes, 2:1-23.

Clark, H. H., and Schaefer, E. F. (1989). Contributing to discourse. Cognitive Science, 13:259-294.

Clark, H H., and Wilkes-Gibbs, D. (1986). Referring as a collaborative process. Cognition, 22:1-39, 1986.

Hulteen, E. A. (1991). User-centered design and voice-interactive applications. Proceedings, AVIOS '91: Voice I/O Systems Applications Conference, pp. 3-8.

Norman, D. A. and Draper, S. W. (1986). User centered system design. Erlbaum, Hillsdale, NJ.

Payne, S.J. (1990). Looking HCI in the I. Human-Computer Interaction INTERACT '90, Elsevier Science Publishers B.V., North Holland.

Schmandt, C. M. and Hulteen, E. A. (1982). The intelligent voice- interactive interface. Proceedings, Human Factors in Computing Systems, 363-366.

Yngve, V. H. (1970). On getting a word in edgewise. Papers from the sixth regional meeting of Chicago Linguistic Society, 567-578.