Problem-Sensitive Response Generation in Human-Robot Dialogs

Petra Gieselmann*
Universität Karlsruhe, Germany
petra@ira.uka.de

Mari Ostendorf
University of Washington, Seattle USA
mo@ee.washington.edu

Abstract

We develop a new mechanism to detect and respond to miscommunications in human-robot dialogs, distinguishing between computer misunderstandings vs. human inexperience. Problem indicators drive an error/help state machine, which augments the dialog state and is used in tailoring response generation. A user study shows that the task success rate and user satisfaction is improved substantially by the two-part miscommunication model.

1 Introduction

A major challenge in spoken dialog systems is miscommunication: speech recognition errors and misunderstandings often result in error spirals from which the user can hardly escape. This leads to user frustration and task failure. The problem of error handling has been the subject of several studies, and errors often cause system designers to use a dialog management strategy based on system initiative and explicit confirmation. In human-robot dialogs, miscommunication can arise not only from imperfect speech recognition and understanding, but also from user uncertainty about the robot's knowledge and capabilities. Moreover, in most applications for humanoid robots, there is a need for a mixed-initiative dialog strategy to support more natural communication.

To handle human-robot miscommunication in this context, we have developed a strategy to detect and track two types of problems: computer misunderstanding vs. user need for help. Computer misunderstandings may be due to speech recognition or understanding errors. Users may need help, either because they are unsure how to proceed in a dialog or because they are asking for something that is beyond the current capability/knowledge of the robot. From the robot's perspective, these problems can sometimes be hard to distinguish, but they require different responses in order to resolve miscommunications.

Consider the following extract of an actual dialog with our robot, where "U" is the human user and "R" is the robot. The robot knows about things in the kitchen; it can help with meals by reciting recipes and bringing things to the human; but it cannot actually cook.

U: Robbi, I am very hungry. Could you please prepare something to eat?
R: In the fridge, there are tomatoes, eggs, milk, and butter. With these ingredients you can make different dishes, such as Spaghetti Napoli or a pancake. What do you want to eat?
U: Spaghetti Napoli please.
R: Sorry, I didn't catch that.
U: Robbi, you just told me that it is possible to cook some Spaghetti Napoli. I would like to have that please.
R: Okay.
U: Fine, thank you.
R: Thank you.
U: Could you prepare something, a dessert?
R: [silence]
. . .
U: Don't you want to cooperate with me?

In the first utterance, the robot failed to detect that "prepare" indicated that the user was asking it to do something beyond its capability. Thus, the user assumes that it can cook, which seems to be confirmed when the robot says "okay" to "it is possible to cook...," though for the robot this means that the ingredients are available. Recognition errors further complicate the misunderstanding, and not surprisingly lead to frustration.

To deal with such problems, variables indicative of different types of errors are tracked, based on analyses of the recognized user utterance. Together with the current dialog state and "problem state," these variables are used to predict whether the conversation is functioning normally vs. in an error spiral or help-needed condition. The response generation strategy is then adjusted accordingly, both in terms of the type of response and its wording. The details of this strategy and experimental validation are described below, following a review of related work.

2 Related Work

There have been several analyses of communication failures in human-computer dialog, looking at characteristics of utterances where speech understanding errors occur, as well as those of attempted corrections of errors. Studies have found that the speech recognition error rate increases with increasing depth into the error correction subdialog (Swerts et al., 2000), as does user frustration

*Now with Lucy Software and Services GmbH, Munich, Germany

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 219–222,
Antwerp, September 2007. ©2007 Association for Computational Linguistics

(Bulyko et al., 2005). There are studies showing that error corrections have acoustic and prosodic features different from normal user utterances (Swerts et al., 2000), and combining acoustic and lexical cues to detect corrections, e.g. (Kirchhoff, 2001). Such studies inform speech recognizer design as well as automatic error (correction) detection.

Other studies have focused on factors that impact the dialog management strategy. Shin et al. (Shin et al., 2002) analyzed 161 dialogs from the NIST 2000 Communicator Evaluation (Walker et al., 2001) in terms of system behavior in order to find out how users discover that an error occurred. The results revealed the need for more explicit confirmations, since users need more time to get back on track and fail more often when they discover errors through implicit (vs. explicit) confirmations. Results from human-human communication also stress the need for explicit confirmations in error subdialogs (Gieselmann, 2006). An approach for using error correction detection output to decide between different degrees of system initiative in the generation strategy is outlined in (Bulyko et al., 2005), together with generation wording variations motivated by studies showing effects on user frustration. The goal of this work is to extend the results on dialog strategy and response wording to problems that include not only error handling but also human inexperience, with the goal of shortening miscommunications and increasing user satisfaction.

Within the robotics community, new application domains such as taking care of old people, delivering hospital meals, etc., are driving the development of robots that can interact with humans. It is considered important that people can communicate with these robots as to another human (Sidner and Dzikovska, 2005). Most research in this field concentrates on designing the robot as similar as possible to a human in terms of both its appearance and its communicative behavior (Breazeal, 1999). The focus here on help responses is consistent with this view.

3 Task and Baseline System

Our robot can accomplish different tasks in the household environment; e.g. it can deliver and retrieve kitchen items, switch on or off lights, and provide information about recipes or about the contents of the refrigerator (Gieselmann et al., 2003). The robot should be able to interact with inexperienced and older users, e.g. in assisted living situations, so it is important that the communication be as comfortable as possible for the user. In addition, since the robot does not yet have all of the capabilities of a human and an inexperienced user will not know its limits, it is important that the robot can inform the user about its capabilities.

For speech recognition, we use the JANUS Recognition Toolkit with the IBIS decoder which decodes using a grammar controlled by the dialog manager, which penalizes specific rules depending on the situational context (Fügen et al., 2004). The recognizer grammar also provides a parse for interpreting the utterance. It is a context-free grammar enhanced by information from the ontology defining all the objects, tasks and properties about which the user can talk. The parse tree is converted into a semantic representation and added to the current discourse. The semantic representation consists of the speech act and the objects/properties expressed within the user utterance.

For dialog management, we use the TAPAS dialog tools (Holzapfel, 2005) based on the language- and domain-independent dialog manager ARIADNE (Denecke, 2002), which uses typed feature structures to represent semantic input and discourse information. If all the information necessary to accomplish a goal is available in discourse, the dialog system calls the corresponding service. Otherwise, clarification questions are generated using a template-based approach.

4 Mixed Initiative Dialog Management

Our strategy is to try distinguish between problems due to system errors vs. human inexperience, using different indicators of possible communication problems and a separate problem state model with problem-sensitive response generation, as described next.

4.1 Factors Indicating Problematic Situations

Computer misunderstandings can occur for a variety of reasons. The system has to cope with high variability in spontaneous speech, self corrections, segmentation errors, and barge-in, for example. An utterance may include words that are out of the recognizer's vocabulary, either an infrequent wording of a known concept or a totally new concept. Since the recognizer will hypothesize words that are consistent with its vocabulary and language model, the robot can only detect these problems indirectly. Implicit error indicators we use include:

- *The utterance is not parsed or only partly parsed.*
- *No speech act can be found, neither in the user utterance nor in the discourse.*
- *The user utterance is inconsistent with the current discourse or with the robot's expectations.*
- *The user repeatedly asks for the same information.*

In addition, some problems are explicitly indicated:

- *The user explicitly asks for help.*
- *The user tries to correct a preceding utterance.*
- *The user asks for something that the robot knows it cannot yet do, such as cleaning.*

4.2 Problem State Model

For representing different problems, we developed a 4-state finite-state automaton on top of the dialog manager:

- **Start state**: Used at the start of a dialog and between tasks as an idle state; the discourse history is empty.
- **Error state**: Information needs to be corrected.
- **Help state**: The user does not know how to proceed and needs help by the robot about its capabilities.
- **Normal state**: No known problematic situation.

The transitions between the states are rule-based, determined by the information in the discourse history and the user utterances, and the problem indicators. The robot is initially in the start state and goes to the normal state as long as no problems occur. Implicit problem indicators trigger a transition to either error state or help state, depending on whether there is information available in the discourse that the user might want to correct. The user stays in the help state (or in the error state) as long as the problems persist. After a non-problematic utterance the user returns to the normal state. To switch from the help state to the error state, a user utterance must contain some information which is put in the discourse. The user can also put the system into the help state or the error state directly by uttering an explicit help request or error correction, respectively. In addition, whenever a user asks for a known task the robot cannot accomplish, such as cleaning, the user is also transferred to the help state. When an error is resolved, the user goes back to the normal state. The system goes back to the start state and the discourse is cleared whenever a user request to the robot has been met or the user explicitly clears the discourse using an utterance such as ”start over” or ”abort”.

4.3 Problem-Sensitive Response Generation

In order to appropriately respond to the user, we have the following features to keep track of the ongoing situation:

- HELP NECESSITY: a variable that increases with each problem indicator, and decreases with a transition to the normal state (to some minimum).
- ERROR SPIRAL: count of the number of successive turns in the error state, cleared after a transition to the normal or start states.
- USER KNOWLEDGE: a list that contains the information already given to the user and how many of times it was given.[1]

Within the help state, the user will get information about the robot's capabilities. The full set of robot capabilities is too large to describe in one response, so we use a set if responses organized according to a task hierarchy. At the highest level, the most general capabilities are described, i.e. for a dialog with a new user, and details related to those capabilities are covered in lower level responses. The user knowledge list is used to determine whether the user has already been given a particular help message. If so, the user is either given a different help message for that dialog state or the robot asks the user if s/he would like to hear again about the robot capabilities. When the help necessity gets above a given threshold, the robot asks the user to speak some predefined sentences to better adapt to the user's voice, and the problem state is reset to the "start" state.

	Baseline	V1	V2
No Predefined Task			
Concept Error Rate	68%	52%	49%
No. of new Concepts	5.0	2.8	4.6
With Predefined Task			
Concept Error Rate	50%	42%	25%
No. of new Concepts	3.0	1.4	2.1
Task Completion Rate	57%	70%	96%
Turns per Task	8.4	5.1	2.7

Table 1: Results of the User Study

Within the error state at the beginning of the dialog, the user is asked to check microphone placement. Later, potential errors are handled by a repeat request, with different wording depending on the error spiral as in (Bulyko et al., 2005). In cases of repeated requests that are out of scope, the robot explicitly tells the user tasks that it cannot do.

5 Experimental Details and Results

We conducted a user study to assess the impact of using a general help/error state vs. separating the help and error correction modes. Two different development cycles of the dialog system were tested and compared to a baseline system that had no explicit error handling. Version 1 (V1) used a dialog management and generation strategy with a single state for errors and help together, and version 2 (V2) includes a division of the problem handling into error vs. help states.

We tested V1 and V2 each with 8 users, with no overlap of people in these groups. The baseline system was tested with 3 trials, with 1 person running two trials of the baseline system and 1 trial of V1. Of the 16 people participating, half were native speakers of English and half were fluent English speakers with another native language. All subjects were familiar with computers, but only six had talked to a dialog system before.

The user study consists of three parts. The first part was a free interaction with the robot: users were told that

[1] We track only the current interaction; long term knowledge from multiple interactions is not addressed here.

they had a new household robot that can support them in the kitchen. This situation is more realistic, but also harder for the users because they have limited knowledge of what the robot can do. In the second part, each user was given (the same) 10 tasks to accomplish with the robot. Using specified tasks, we can assess task completion, but we get less information on the types of capabilities that users expect. After the dialog with the robot, users fill in a short questionnaire. They answered three directed questions about how much they liked the system, how successful they were, and how much they would like to use such a robot again. In additional open questions, participants could report their problems and suggestions for further improvements.

To evaluate the dialogs, we measured concept error rate (percent of semantic concepts not understood by the system for whatever reason) and tracked the number of new concepts introduced. The semantic concepts include actions (e.g. bring, report) and objects (e.g. cup, cabinet) in the robot's ontology. For the predefined tasks, we also tracked task completion and number of turns per task.

The fact that the concept error rate decreases with each design cycle (cf. Table 5) confirms the usefulness of error handling in general, and specifically the separation of error and help needs. As expected, the concept error rate and the average number of new concepts decrease when the subjects are given predefined tasks. Note that, even when the tasks are predefined, users still invent new concepts that the robot does not know, so the help functionality is still useful. (The drop in the number of new concepts between the baseline and V1 may be due to user learning; all users of V2 were new to the task.) For the dialogs with predefined tasks, there is an increase in the task completion rate and a decrease in the number of turns per task for each step in the design cycle, with bigger changes in moving from V1 to V2. Differences found in the condition without predefined tasks do not reach statistical significance, but all the differences within the predefined task condition are significant (*p*-value smaller than .008 for concept error rate and *p*-value smaller than .005 for turns per tasks and task completion rate).

The results of the user survey revealed that in V2 the users liked the robot more and felt they had been more successful in their interactions, compared to V1. The differences in responses related to whether they would like to use such a robot again were not significant, possibly because problem handling does not impact the robot's actual capabilities. Within the free-text answers, some users mentioned the nice recovery after misunderstandings and stressed that it was very clear about its capabilities.

6 Conclusion

In summary, we developed a new dialog management strategy which is sensitive to different types of miscommunications in human-robot dialogs. We use several types of problem indicators to drive state transitions in a 4-state indicator of error/help modes. The generation strategy is modified according to the type of problem, if any. The results of a user study showed that the task success rate, concept accuracy, and user satisfaction are improved substantially by these changes.

In the future, the error handling component can be improved by expanding the problem state space, and including new features such as word confidence, out-of-vocabulary word detection, acoustic cues, and new problem indicators. Another potential direction is to use the problem indicators as input to a Markov decision process for controlling the dialog state. Finally, we note that automatic learning of new concepts and skills on the robot's side will require dynamic update of the problem tracking and help response generation mechanisms.

Acknowledgment: This research is partially funded by the German Research Foundation (DFG) under SFB 588.

References

C. Breazeal. 1999. Robot in society: Friend or appliance? *Proc. Workshop on Emotion-based Agent Architectures*.

I. Bulyko *et al.* 2005. Error correction detection and response generation in a spoken language dialogue system. *Speech Communication*, 45:271-288.

M. Denecke. 2002. Rapid prototyping for spoken dialogue systems. *Proc. ACL*, pp. 1-7.

C. Fügen, H. Holzapfel, and A. Waibel. 2004. Tight coupling of speech recognition and dialog management. *Proc. ICSLP*.

P. Gieselmann *et al.* 2003. Towards multimodal communication with a household robot. *Proc. HUMANOIDS*.

P. Gieselmann. 2006. Comparing error-handling strategies in human-human and human-robot dialogues. *Proc. KONVENS*, pp. 24-31.

H. Holzapfel. 2005. Towards development of multilingual spoken dialogue systems. *Proc. LTC*.

K. Kirchhoff. 2001. A comparison of classification techniques for the automatic detection of error corrections in human-computer dialogues. *Proc. NAACL Workshop on Adaptation in Dialogue Systems*.

J. Shin *et al.* 2002. Analysis of user behavior under error conditions in spoken dialogs. *Proc. ICSLP*, pp. 2069-2072.

C. Sidner and M. Dzikovska. 2005. A first experiment in engagement for human-robot interaction in hosting activities. N.O. Bernsen *et al.* (Eds.), *Advances in Natural Multimodal Dialogue Systems*.

M. Swerts, J. Hirschberg, and D. Litman. 2000. Corrections in spoken dialogue systems. *Proc. ICSLP*.

M. Walker *et al.* 2001. DARPA Communicator dialog travel planning systems: the June 2000 data collection. *Proc. Eurospeech*, 2:1371-1374.