Dialogue Patterns and Misunderstandings

John Aberdeen and Lisa Ferro

The MITRE Corporation
Bedford, MA 01730 USA
{aberdeen,lferro}@mitre.org

Abstract

Disruptive errors are common in many human-computer (HC) dialogues. We manually applied initiative and dialogue act annotations to HC dialogues in the travel domain in an effort to find patterns that are predictive of misunderstandings. While we found some interesting patterns of dialogue acts, we also found that a detailed understanding of the misunderstandings in our data required us to perform more in-depth analysis than is possible just by examining dialogue acts. Our hope is that analyses such as these will inform the design of HC dialogue systems, so that systems may predict problematic situations in order to deal with them more effectively.

1. Introduction

Communication problems are common in spoken language dialogue systems, and can arise for many different reasons (Dybkjær and Bernsen, 2002). In previous work we attempted to find misunderstandings through a combination of manually applied semantic annotation tailored to the dialogue domain (air-travel reservations), and an automatic algorithm to examine the annotated dialogues (Aberdeen et al., 2001). Manual annotation is time and labor intensive, and while we are committed to the detailed understanding that one can gain from it, we should strive to minimize it.

There are at least two methods that can be used to minimize manual annotation. One is to simply automate annotation wherever possible. Walker and Passonneau (2001) developed a method to automatically add dialogue act tags to the system side of human-computer (HC) dialogues in the travel domain, using a pattern matching algorithm. This is a very efficient method for reducing manual annotation (since there is none!), and it provides valuable insights about the system side of HC dialogues. This method also makes it possible to annotate very large amounts of dialogue data quickly.

A second method to reduce manual annotation is to reuse existing annotations. Here we attempt to discover what can be learned about misunderstandings from general-purpose annotations that are not specific to any particular dialogue domain. Specifically, we are investigating what we can learn about misunderstandings from dialogues annotated with dialogue act and initiative tags. Our annotations are manually applied, so it relatively easy to annotate both the system and user sides of HC dialogues. This is in contrast to automatically applied annotations, which can be difficult to apply to the user side of dialogues, due to the greater variability of user utterances.

2. Data and Annotations

The subject of our analyses was a subset of dialogues collected during the 2001 six-month data collection conducted by the DARPA Communicator program (Walker et al., 2002). These are dialogues between paid subjects and research prototype air-travel reservation systems. After each call, subjects were requested to fill out a short questionnaire (five questions, each on a five-point Likert scale) to assess their satisfaction (Walker et al, 2002). We applied our annotations to a total of 80 dialogues, ten dialogues for each of eight participating systems.

2.1. Initiative Annotation

We applied Walker and Whittaker's (1990) approach to initiative tagging. Each turn was tagged with which participant has control at the end of that turn, based on the utterance type. Below we list the rules for tagging each utterance type; a PROMPT is an utterance that does not express propositional content, such as *Yeah*, *Okay*, *Uh-huh*, etc. The classification refers to the illocutionary force of the item, rather than to its particular syntactic form.

ASSERTION: speaker has initiative unless it is a response to a question or command
QUESTION: speaker has initiative unless it is a response to a question or command
COMMAND: speaker has initiative
PROMPT: hearer has initiative

Two annotators tagged each utterance with a USER-INITIATIVE or EXPERT-INITIATIVE tag based on their characterization of the utterance type (EXPERT corresponds to SYSTEM in HC dialogues). Overall interannotator agreement was 0.94, but the kappa score was 0.5. We believe that the reason for the low kappa score is that with an extremely small tag set such as this one (two tags), any disagreement is magnified by the kappa metric; an extraordinary level of agreement is required to obtain a reasonable kappa score in this instance. Nevertheless, with our low kappa score, we are uncomfortable drawing conclusions about misunderstandings from initiative annotations.

2.2. Dialogue Act Annotation

For DA annotation we used a modified subset of the CSTAR tag set, documented in Doran et al. (2001), and shown below.

ACCEPT: *that sounds great* (of an offer)
ACKNOWLEDGE: *okay* (backchannel)
AFFIRM: *yea* (an answer to a question)
APOLOGIZE: *i'm confused*
DEMAND-CONV-INFO: *please say yes or no*
DEMAND-SIT-INFO: *please enter your personal identification number followed by the pound key*
DEMAND-TASK-INFO: *and the one before that*
GIVE-SIT-INFO: *this call is being recorded for system development you may hang up or ask for help at any time*
GIVE-TASK-INFO: *here we've got you on american flight nine thirty eight*

NEGATE: *no* (an answer to a question)
NOT-UNDERSTAND: *i'm not sure what you said*
OFFER: *so on the twentieth you want me to look at the return*
OPEN-CLOSE: *you're all set then*
PLEASE-WAIT: *hold on while i check availability*
REJECT: *no i don't need a car* (of an offer)
REQ-CONV-ACTION: *could you repeat it please*?
REQ-SIT-ACTION: *please restrict your requests to air travel*
REQ-SIT-INFO: *how do you spell his last name*?
REQ-TASK-ACTION: *you'll have to check that one too*
REQ-TASK-INFO: *what date will you be traveling*?
SUGGEST-CONV-ACTION: *try saying a short sentence*
THANK: *thank you for using our communicator air travel system*
VERIFY-CONV-ACTION: *did i make that clear*?
VERIFY-TASK-ACTION: *and that's all set*
VERIFY-TASK-INFO: *from paris to denver on tuesday july 18*
YOURE-WELCOME: *you're quite welcome*

We attached a single tag to each utterance that contained some speech, i.e. was not composed entirely of non-speech annotation like *pause* or [click]. Because there could be multiple utterances in a turn there were often multiple dialogue acts (DAs) per turn. Where there were multiple sequential DAs of the same type, we collapsed them under a single tag on the assumption that they were combining to "perform" that DA. Two annotators tagged the dialogues with DA tags, and achieved an overall interannotator agreement of 0.82, and a kappa score of 0.8.

3. Results

We found that several types of dialogue acts were correlated with user satisfaction. Due to the sparse use of some of the tags in our DA set, we do not have sufficient numbers to calculate correlations for all DAs. Table 1 shows correlations with user satisfaction greater than 0.5 in either direction, and for which we had at least 30 instances of the tag.

Table 1: Correlations between DAs and User Satisfaction

Dialogue Act	Correlation w/User Satisfaction
APOLOGY (system)	-0.59
DEMAND-TASK-INFO (system)	-0.59
NEGATE (user)	-0.51
REJECT (user)	-0.56

We were not surprised to find that the DAs apologize, not-understand, negate, and reject were negatively correlated with user satisfaction. We also found that demand-task-info (e.g., "state your departure city") was negatively correlated with user satisfaction (as distinct from request-task-info, e.g., "where are you flying from"). This may be a reflection of a dialogue style employed by many systems, in which a request-task-info utterance is used initially, and only if that does not result in a slot fill is a more forceful demand-task-info utterance used. We did not find any correlations between initiative patterns and user satisfaction.

Based on these results, we were disappointed to find that deep insights about misunderstandings were not readily obtainable from our DA and initiative annotations. Thus, we found it necessary to perform much more detailed analyses of the dialogues.

4. Detailed Analysis

The designers of mixed-initiative dialogue systems expect users to be aware when communication with the system has gone awry, and to cooperate in correcting such misunderstandings. One common design feature that aids users in this task is the *implicit confirmation*, which echoes back information from the previous turn while continuing to gather information:

(1)
2 System: okay from hartford to orlando. what date will you be travelling?
3 User: leaving hartford on october the thirty first
4 System: okay, from hartford to orlando on wednesday october 24. can you provide the approximate departure time or airline?
5 User: i need to leave on october thirty first

In utterance 5, the user interjects a correction to the implicit confirmation. Much valuable research has been conducted to explore methods by which systems can accurately detect when misunderstandings occur (e.g., Krahmer et al. (2001), Litman et al. (1999), Lendvai et al. (2002), Walker et al. (2000) and Van den Bosch et al. (2001)). To complement that research, our goal in this research task was to develop an approach to corpus analysis for exploring three areas: (a) What cues are available for *users* to spot misunderstandings? (b) What recovery mechanisms exist? and (c) How effective are these cues and recovery mechanisms in resolving the problem? By having a clearer understanding of the dialogue from the user's perspective, system designers can build systems that better support users in their attempts to correct misunderstandings. In this section we report on the early stages of this effort, presenting the corpus analysis scheme as it was applied to a small set of data.

4.1. Methodology

A single annotator manually inspected 40 of the dialogues (five for each of eight participating systems) and identified 137 misunderstandings caused by the system. For each misunderstanding detected, four features were identified:

1. The *type of error* that apparently led to the misunderstanding. Because we only have access to the utterances themselves, this label is approximate. However, it does serve to distinguish major groupings of error types, such as ASR errors vs. those caused by the inability of the system to adapt to the user's expectations.

2. The *surface evidence* available to the user that a misunderstanding had occurred, such as the system repeating a question that the user had already answered. These indicators are not to be confused with "aware sites" (Litman et al. 2001), which identify *user* utterances that reveal the user's awareness of a misunderstanding.

Table 2: Categories of Evidence Tabulated by Outcome

Evidence	Outcome			Total
	Resolved	Unresolved	Unknown	
assumes unverified facts		**1**		**1**
explicit confirmation	2	5		7
failure to obey command		**10**		**10**
gap in normal script		**4**		**4**
ignores topic switch		**3**		**3**
implicit confirmation	9	18		27
implicit confirmation AND explicit confirmation		1		1
implicit confirmation AND repeated prompt	1	1		2
implicit confirmation AND requests clarification	1			1
repeated offer	1	2		3
repeated prompt	17	8		25
repeated prompt AND explicit confirmation	1			1
repeated prompt AND ignores user correction		1		1
requests clarification	3	2		5
response is contradictory	**1**			**1**
response references non-topic		**1**		**1**
system alert	2		1	3
system alert AND failure to obey command	2	3		5
system alert AND repeated prompt	14	21	1	36
Total	54	81	2	137

3. The *correction mechanism* used by either party to recover from the misunderstanding, which in extreme cases can include the user issuing a "start over" command.

4. The *outcome* of the misunderstanding: either Resolved, Unresolved, or Unknown (for two cases in which the dialogue mysteriously ends). A misunderstanding was considered Resolved if was corrected by the end of the next pair of user-system turns. If an outcome was Unresolved because the misunderstanding re-occurred or persisted, it was counted as a new misunderstanding.

Example (2) shows a sample exchange and the resulting markup, and is then followed by a discussion of the overall results.

(2)

0 System: what city are you flying to?
1 User: new york
2 System: sorry, i didn't understand that. what city are you flying to?
3 User: new york
4 System: flying to new york..

Error type: ASR
Evidence: system alert AND repeated prompt
Correction Mechanism: repeated prompt
Outcome: Resolved. (Repeated information is understood.)

4.2. Results

4.2.1. How does the user know something has gone wrong?

We observed 13 different indicators of misunderstandings, and often more than one would be present in a given turn. As can be seen in Table 2, most of the indicators are those that are explicitly designed to expose misunderstandings; explicit confirmations, implicit confirmations, the system requesting a clarification, repeated prompts and offers, and system alerts (e.g., "*I'm sorry. I didn't understand you*") correspond to 85 percent of the misunderstandings in this small data set. These indicators are typically associated with failed attempts to fill slots, as seen in examples (1) and (2), and make it very clear to users (if they're paying attention) that the system has misunderstood the user.

But there are other types of errors and thus other types of evidence of misunderstandings present in HC dialogues. These other indicators, which we have highlighted in bold type in the table, are *assumes unverified facts*, *failure to obey command*, *gap in normal script*, *ignores topic switch*, *response is contradictory*, and *response references non-topic*, which combined correspond to 15 percent of the misunderstandings. We explore these in more detail here. We looked at the apparent error types associated with these bold-typed indicators and found that 65 percent of the instances are associated with what we termed "script errors." (Compared to the data set as whole, in which 73 percent of the misunderstandings are associated with ASR or suspected ASR errors). These script errors come in two forms. In one form, the user tries to take the conversation in a direction that the system is not prepared to handle. If the system fails to respond, this results in the *failure to obey command* indicator shown in Table 2. For example, in (3), the system is expecting a *yes* or *no* answer, and apparently nothing else. The user's "start over" command is ignored and the system proceeds to look up flight information in the database.

Table 3: Error Correction Mechanisms Tabulated by Outcome

	Correction Mechanism	Outcome			Total
		Resolved	Unresolved	Unknown	
System Initiated	implicit confirmation AND repeated prompt		1		1
	repeated offer	1			1
	repeated prompt	28	27	1	56
	repeated prompt AND user rephrases info	2			2
	system requests clarification	3	2		5
	system requests verification	3	6		9
User Initiated	user hangs up		2		2
	user interjects correction	8	10		18
	user issues cancel command	4	2		6
	user issues start over command		6		6
	user repeats command	2	7		9
	user repeats info	1			1
	user rephrases info	1			1
	user rewords query	1			1
	unknown			1	1
	none		18		18
	Total	54	81	2	137

(3)

10 System: are you going back to salt lake city? say yes or no please.
11 User: no i'd like to start over
12 System: okay, it will just take a moment while i look…

The other form of script error, which is less common, is when the system deviates from the normal script for no apparent reason. In (4), which contains two consecutive misunderstandings, the system is deviating from its normal procedure of verifying information before adding a flight to the itinerary, and then further deviates by jumping right to the pricing stage without finishing the return flight reservation.

(4)

14 System: i couldn't find any flights on american. delta flight 759 connecting to delta flight 2034 in atlanta has been added to your itinerary. you depart melbourne florida at 7:00 am and arrive in norfolk at 11:15 am. what date will you be returning on?
15 User: twenty fifth
16 System: would you like me to find out how much your trip will cost?

The *failure to obey command* indicator is also often due to apparent ASR errors; e.g., the user asks for the next flight option and the system instead repeats the previous flight option. What characterizes all these misunderstandings is that the system seems oblivious – oblivious to the user's wishes, oblivious to the possibility of a misunderstanding, and oblivious to the user's attempt to correct the misunderstanding. Unlike those situations in which system is designed to seek verification, these types of misunderstandings are difficult to recover from, as discussed in the next section.

4.2.2. What mechanisms exist for the system and/or user to recover from a misunderstanding?

Table 3 shows the five system-initiated correction mechanisms identified and the eight user-initiated correction mechanisms we observed; again sometimes more than one mechanism is used. Some types of indicators of misunderstandings seen earlier in Table 2 are also mechanisms for repairing the misunderstanding. When the system fails to understand an answer to its question, simply repeating the question ("repeated prompt") can be quite effective. But 18 (13 percent) of the misunderstandings have no correction mechanism associated with them (labeled "none"), and it is not surprising that the outcome is always Unresolved in such cases. We examined the overlap between these cases with no correction mechanism and the "system seems oblivious" errors discussed earlier, and found that 50 percent of such errors have no correction mechanism. In the remaining 50 percent, users repeated their command (which is next most common response to these types of misunderstandings), tried to interject a correction, or issued a "start over" command. The latter could get the user into a frustrating loop if it was the "start over" command that the system was ignoring, as was the case in (3).

4.2.3. How effective are these indicators and recovery mechanisms?

In the preceding tables we tabulated the results by *Outcome* in order to show the general direction in which the data was leaning. However, we urge caution in drawing conclusions from these numbers. Not only is the data set very small, it encompasses eight different systems. What works for one system may not work well for another. For example, in Table 3 it appears that "user interjects correction" mechanism is successful less than half the time. However, this is in large part due to different systems having different abilities in being able to cope successfully with such interruptions. Thus, this analysis methodology

needs to be applied to larger amounts of data from single systems. As said earlier, our purpose in this paper is mainly to illustrate the analysis scheme.

However, in spite of the small data size, the patterns that are emerging do reveal a potential trouble spot that warrants further attention by system designers, namely, the crossover between the bold-typed indicators in Table 2 and the lack of effective correction mechanisms for them. It must be noted that the systems in this corpus were research systems, and thus may have lacked some recovery mechanisms found in deployed systems. Real dialogue systems will often switch the user to a human operator if the system repeatedly fails; the lack of such recourse in the research setting may have led some errors to persist longer than they normally would. Nevertheless, this still implies that this trouble spot is likely in need of such rescue measures.

4.3. Planned Enhancements and Future Work

In the future, we want to perform this analysis on more data, and with more than one annotator so we can measure interannotator agreement. The time has also come to settle on a fixed set of tags, rather than generating an open-ended set of descriptors like we did in this preliminary investigation. It would also be useful to track the number of turns that occur between the initial evidence of a particular misunderstanding and the time that it is resolved; similarly, to track the number of turns spent on attempting to resolve misunderstandings that are never resolved. To do this, we need to add an additional layer of information – tracking how long it takes for each participant to obtain the information being sought (e.g., how many turns it takes for the system to correctly understand the departure airport). Finally, since the goal of this task is to explore the user's view of misunderstandings, the results should be correlated with user satisfaction questionnaires for the same dialogues.

5. Conclusions

In this work we intended to explore the possibility that misunderstandings might be detected and understood through the analysis of manually-applied, general-purpose annotations. We noticed some interesting correlations between certain dialogue acts and user satisfaction. Based on the small amount of data that we annotated we are unable to draw conclusions about misunderstandings from patterns of initiative and dialogue acts. However, a much more detailed analysis of misunderstandings yielded several insights about the source and nature of misunderstandings, as well as the correction mechanisms employed by users and systems. Detailed analyses such as this can lead to improvements in future spoken language dialogue systems.

6. Acknowledgements

The authors would like to thank Christine Doran and Janet Hitzeman for assistance with annotation.

7. References

[1] Aberdeen, J., Doran, C., Damianos, L., Bayer, S., and Hirschman, L., "Finding errors automatically in semantically tagged dialogues", *Proceedings of the First International Conference on Human Language Technology Research*, p. 124-128, 2001.

[2] Doran, C., Aberdeen, J., Damianos, L., and Hirschman, L. "Comparing several aspects of human-computer and human-human dialogues." *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue, Aalborg, Denmark*, 2001.

[3] Dybkjær, L., and Bersen, N.O. "Tagging communication problems in spoken dialogue systems: on-line or off-line?" *Proceedings of the ISLE Workshop on Dialogue Tagging for Multi-Modal Human Computer Interaction, Edinburgh, Scotland*, 2002.

[4] Krahmer, E., Swerts, M., Theune, M., and M.E. Weegels, "Error detection in spoken human-machine interaction", *International Journal of Speech Technology,* 4(1):19-30, 2001.

[5] Litman, D.J., Walker, M.A., and M.S. Kearns, "Automatic detection of poor speech recognition at the dialogue level", *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (ACL'99), p. 309-316, 1999.

[6] Litman, D.J., Hirschberg, J., and M. Swerts, "Predicting user reactions to system error", *Proceedings of the 39th Meeting of the Association for Computational Linguistics* (ACL'01), 2001.

[7] Lendvai P., Van den Bosch A., Krahmer E., and M. Swerts, "Improving machine-learned detection of miscommunications in human-machine dialogues through informed data splitting", *Proceedings of the ESSLLI 2002 Workshop on Machine Learning Approaches in Computational Linguistics*, 2002.

[8] Walker, M., Rudnicky, A., Prasad, R., Aberdeen, J., Bratt, E.O., Garofolo, J., Hastie, H., Le, A., Pellom, B., Potamianos, A., Passonneau, R., Roukos, S., Sanders, G., Seneff, S., and Stallard, D. "DARPA Communicator: cross-system results for the 2001 evaluation." *Proceedings of the International Conference on Spoken Language Processing, Denver, CO, USA*, 2002.

[9] Walker, M. and Passonneau, R. "DATE: a dialogue act tagging scheme for evaluation of spoken dialogue systems", *Proceedings of the First International Conference on Human Language Technology Research*, p. 66-73, 2001.

[10] Walker, M., Wright, J., and I. Langkilde, "Using natural language processing and discourse features to identify understanding errors in a spoken dialogue system", *Proceedings of the 17th International Conference on Machine Learning*, 2000.

[11] Walker, M. and Whittaker, S. "Mixed initiative in dialogue: an investigation into discourse segmentation." *Proceedings of the 28th Meeting of the Association for Computational Linguistics*, 1990.

[12] Van den Bosch, A., Krahmer, E., and M. Swertz, "Detecting problematic turns in human machine interaction: rule-induction versus memory-based learning approaches", *Proceedings of the 39th Meeting of the Association for Computational Linguistics* (ACL'01), p. 362-369, 2001.