

Clarification Questions to Improve Dialogue Flow and Speech Recognition in Spoken Dialogue Systems

Ulf Krum, Hartwig Holzapfel and Alex Waibel

Interactive Systems Labs
Universität Karlsruhe

{krum, hartwig, waibel}@ira.uka.de

Abstract

Within human-machine conversation, clarification is vital and may consist of various forms, as it is may be due to many different effects on different levels of communication. In this paper, we present a strategy for detecting situations where a need for clarification exists in a natural spoken dialogue system. We define rule sets which enable us, via an anomaly analysis, to detect these critical situations. Through the use of such rule sets, we show that it is possible to enhance the strategy in such a manner that more different situations are detected. In a user test, we evaluate the success of the strategy and show that strategies with explicit clarification improve the naturalness of human-machine interaction.

1. Introduction

Within human-machine conversation, Clarification plays a vital role and may take various forms, as it may have different causes at different levels of communication [1, 2, 3].

It is quite obvious that clarification in human-machine conversation is even more important than in human-human interaction. This is because speech recognition is error prone and produces many recognition errors, especially when using distant speech. In addition, systems are not able to interpret semantics and context as humans do and usually lack overall world knowledge. Therefore the dialogue strategy is responsible for the clarification of ambiguous or incomplete information provided by the user.

Earlier we presented the hold strategy, which uses implicit clarification [4]. In this paper we present a strategy for detecting dialogue situations that lead to a context switch. These situations may be caused by the user or by speech recognition errors.

Our approach uses explicitly asked clarification questions after detecting these critical situations. In an analysis, we resolve sequences of dialogue state transitions which indicate such situations. In a small user evaluation we compared the new strategy with the hold strategy, where the new strategy lead to a more natural dialogue flow.

The remainder of this paper is organized as follows: section 2 gives an overview of related work. Section 3 introduces our dialogue management system. In section 4, we discuss causes for clarification and present the ones covered in this paper. In section 5 we describe how we detect the need for clarification in our system. Section 6 presents a catalogue of clarification requests. In section 7 we present a small user evaluation. Finally we conclude our work in section 8.

2. Related Work

The detection of situations where clarification is needed and the appropriate form of the request is a non-trivial challenge.

Schlangen [3] recommends the use of confidences for all guessed hypotheses and over all processing stages of the user input. Depending on the values, the input could be rejected, explicitly confirmed, implicitly confirmed or accepted.

In [2], there are also confidence values used to validate information given in a user utterance. By means of confidence values, the underlying strategy decides whether to accept or reject the user input or to ask partial or alternative clarification questions.

In the speech translation system VERBMOBIL, clarification requests are used in situations where the system has insufficient information to continue processing [5]. These situations concern three aspects: phonological ambiguities, unknown words and semantic inconsistencies. For each aspect an analysis method was developed for the detection of such situations.

In earlier work [4] we presented strategies that lead out of dead end situations in human-robot interaction. In a situation where the recognized input does not fit to the current discourse, the system decides whether to abort the old dialogue, to open a subdialogue or to let the user repeat his utterance. The situation may occur due to errors in speech recognition or be intended by the user. Based on the input confidence, a better fitting hypothesis in the n-best list, and the dialogue state, the system decides which strategy should be applied.

3. Dialogue System Components

For dialogue management we use the TAPAS dialogue framework. TAPAS uses dialogue algorithms developed within the language and domain independent dialogue manager ARIADNE [6] which is specifically tailored for rapid prototyping of spoken dialogue systems. The dialogue manager uses typed feature structures (TFS) [7] to represent semantic input and discourse information. A context-free grammar is used to parse the user utterance. The grammar is enhanced by information from the ontology defining all the objects, tasks and properties about which the user can talk. After parsing, the parse tree is converted into a semantic representation and added to the current discourse. If all necessary information to accomplish a goal is available in discourse, the dialogue system calls the corresponding service. If some information is still missing, the dialogue manager generates questions to request this information.

For speech recognition, we are using the Janus Recognition Toolkit (JRtk) [8] with the Ibis single pass-decoder [9]. We use the option of Ibis to decode with context free grammars (CFG)

instead of statistical n-gram language models (LM). These context free grammars are generated by the dialogue manager that uses the same grammars to convert the resulting parse tree into typed feature structures. In addition, the system offers a tighter integration with Janus by being able to weight (e.g. boost) different grammar rules depending on the dialogue context [10]. The dialogue system uses semantic grammars [11] to interpret

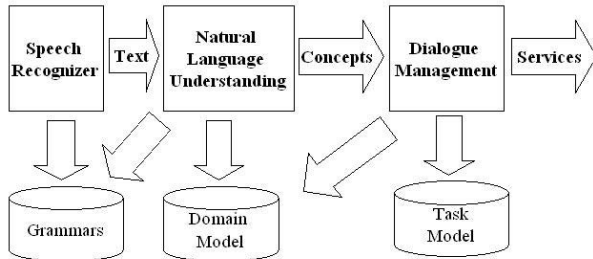


Figure 1: Flow diagram showing the integration of NLU component into the dialogue system.

spoken (or typed) input. The integration of the grammars (natural language understanding) into the system is shown in figure 1. The processing of the dialogue algorithms and the discourse representation are language independent. This allows using general dialogue and discourse algorithms, without depending on language specific peculiarities.

The dialogue manager is task-oriented, and most of the dialogue goals are created to collect information that is required to execute functions defined by the application's API. Additional dialogue goals (and subgoals) allow non task-oriented communication, such as greetings, error correction, etc. More language specific information is required to generate clarification questions or respond to the user.

The dialogue strategy chooses and performs a dialogue move that is most appropriate in the given situation. A move can request new information, generate a clarification question, give information, or generate confirmations. In addition to that it can execute different action, as described above. If a move generates an information request, it needs to describe which information is requested and which type of speech act is used for this request. The system's utterance is created by generation templates.

Clarification dialogues are integrated in the dialogue processing in a manner that affect the updating of the discourse. Each clarification dialogue is controlled by a finite state machine.

4. Causes for Clarification

There are many causes for clarification needs in human-human dialogue. Such as, for example, ambiguous information or acoustical misunderstanding. Human interaction with machines fails even more often than human-human interaction. One reason for this is the fact that speech recognition is not perfect today. Especially if distant speech is used, the input channel is not as clean as for close talk. Various environmental noises appear, caused by reverberation, bad signal to noise ratio, cross talking or other environment noise. These are conditions that impair automated speech recognition.

In our work we focus on clarification requests caused by a

context switch or by utterances which do not fit into the current context.

From the point of view of the dialogue manager, there can be two reasons that lead to situations where the current input does not fit into the dialogue manager's context:

1. the user changes his intention and follows another goal, possibly intending to resume the current goal later; or
2. the speech recognition component produced an error.

The two simplest ways to respond to such a situation are to rely on the output of the speech recognition and open a subdialogue, or to reject the input and ask the user to repeat his answer. It is obviously not satisfying to apply one of the alternatives all the time. On the other hand, the dialogue strategy does not have the ability to decide which alternative may be the preferred. This fact leads to a need for clarification.

As we will present in the next section, we decide whether to ask a clarification request or not based on the history of the dialogue progress including different aspects of the dialogue state. In [2, 3], the input confidences of the actual user input are used to detect needs for clarification. Thus no history is observed. Moreover only one aspect of the dialogue state has been taken into attention: the input confidence.

The hold strategy [4] asks the user its last question if the actual user input will cause the system to switch to another context. Thus implicit clarification is used and the user has to repeat his last input to confirm the context switch or to still follow the old goal.

All systems described above using clarification requests are kind of "active" dialogue managers. The interaction with a human is used to perform tasks that are intended by that human. The speech translation system VERMOBIL provides a kind of "observing" dialogue manager that passively analyzes a dialogue between two humans. If there are situations that require clarification, the dialogue manager activates prior to translating the actual utterance to the opponent. Within a clarification dialogue, insufficient information is completed and used for translation. Insufficient information may cause problems in the translation process, such as unknown words, phonological similarities or missing or inconsistent semantic information.

5. Detecting needs for clarification

Strategies based on TAPAS use abstract dialogue states as a base for decisions for the next move [4].

During the dialogue, the system reaches various dialogue states, each defined by a specific assignment of the variables in the abstract dialogue state. The values of the variables describe certain aspects of the current dialogue situation. With an analysis of the actual and previous states, it is possible to detect anomalies that indicate a need for clarification.

The dialogue state is formally written as:

$d = \langle v_1, v_2, v_3, \dots, v_n \rangle$, where each v_i represents one variable.

The abstract dialogue state used in our clarification strategy contains the following variables:

v_1 : **INTENTION** describes how well the discourse information represents the intention of the user[6]. It is calculated on the basis of the states of dialogue goals.

v_2 : **SELECTEDGOALS** is a set containing all goals that have the state *determined* [6]. This means that the discourse fits these goals.

v_3 : **FINALIZEDGOALS** is a set containing one or none *finalized* goals[6]. This means that there is one goal with state *finalized* and all the information needed for execution is present. Theoretically it is possible to have more than one finalized goal. This is caused by the application description, not by the strategy and should be avoided.

Anomaly analysis is used to detect which transitions from the previous state d_{k-1} to the actual state d_k are critical.

A trigger, as used in our anomaly analysis, is a collection of precepts for the assignment of state variables. It classifies the critical transitions and the corresponding clarification requests. The definition of a clarification dialogue is realized with finite state machines.

The use of explicitly defined triggers allows us to easily expand the anomaly analysis with additional variables of the abstract dialogue state. Furthermore it is possible with this formalism to discover and easily implement new triggers that take into account more critical situations, or to achieve more natural behavior in different situations.

6. A catalogue of clarification requests

In this section we describe some trigger and clarification dialogues. These were implemented in our strategy and tested.

For arbitrary values of the variables we write "–" (don't care).

6.1. Misunderstanding Trigger

If the variable INTENTION holds the value *deselected*, the users utterance did not fit into the current context. This means that there is no goal appropriate to the users input and further that the speech processing component may have produced an error. Therefore a clarification request "*I misunderstood you, please try this once again*" is asked. The trigger is defined as follows:

$$\begin{aligned} d_{k-1} &= \langle -, -, - \rangle \\ d_k &= \langle deselected, -, - \rangle \end{aligned}$$

6.2. Subdialogue Trigger

While the user follows a dialogue goal, the variable switches from *selected* over *determined* to *finalized*, from starting the goal to finalizing it. While the elements in SELECTEDGOALS constitute a subset of each in the previous state. In the case of INTENTION = *finalized*, the variable FINALIZEDGOALS holds a subset of SELECTEDGOALS.

The following assignment causes our strategy to ask the user, if he wants to switch to a subdialogue. Thus it clarifies if the context switch is intended by the user with a appropriate question.

$$\begin{aligned} d_{k-1} &= \langle \{selected|determined\}, G_{s_{k-1}}, \emptyset \rangle \\ d_k &= \langle \{selected|determined|finalized\}, G_{s_k}, G_{f_k} \rangle \end{aligned}$$

$$\text{with } \forall g \in G_{s_k} : g \notin G_{s_{k-1}} \wedge g \notin G_{f_k}$$

An example is given in figure 2, where the variable INTENTION remains unchanged from state d_1 to state d_2 , but the selected goal changed from *SetTable* to *PutSomething*. From the dialogue managers point of view it is obvious that the context of the dialogue has changed. But it can not determine if this was caused by the user or by an error in the speech processing unit.

1	User:	"Please set the table robbi"
	Recognized:	"Please set the table please"
	$d_1 =$	$\langle determined, \{SetTable\}, \{\} \rangle$
	System:	"For how many Persons do you want me to set the table?"
2	User:	"For two persons please"
	Recognized:	"Put two glasses please"
	$d_2 =$	$\langle determined, \{PutSomething\}, \{\} \rangle$
	System:	"I understood that you want me to put glasses somewhere?"

Figure 2: Anomaly classified by Subdialogue Trigger.

1	User:	"Please set the table robbi"
	System:	"For how many Persons do you want me to set the table?"
2	User:	"Bring me a coke."
	System:	"Do you want me to bring you a coke?"
3	User:	"Yes."
	System:	"Here you are. Do you want me to resume setting the table?"
4	User:	"Yes."
	System:	"Which kind of glasses do you want me to put on the table?"

Figure 3: Session with subdialogue.

6.3. Returning from a Subdialogue

In our clarification strategy we implement subdialogues. The current discourse is stored on a stack, when a context switch is intended by the user. The new discourse and dialogue state are computed from the actual user input. There is still a discourse on stack after finalizing the new goal. The dialogue strategy asks the user if he wants to resume the old goal corresponding to that discourse.

In this case we have a triggerless clarification dialogue. This kind of clarification requests are also provided by our strategy and may be invoked whenever the strategy needs it.

Figure 3 shows an example where the user starts a subdialogue. After finalizing the goal of the subdialogue the context of the first goal remains on the stack. The whole session contains two clarification requests. The first one is for clarifying if there is an context switch intended by the user or if the speech processing unit produced an error (Turn 2 and 3). The second one clarifies if the user intended a subdialogue after finalizing the second goal, or if he just canceled the first goal by changing to another (Turn 3 and 4).

7. Evaluation

In order to show that explicit clarification leads to more natural dialogues, we evaluated our strategy within a small data collection. For comparison, we used the previously described hold strategy [4] as a baseline. The data collection was conducted with seven persons. Each person had to complete three different scenarios, each with both dialogue strategies. Together this are 42 Dialogues. Two of the users competed only the first two scenarios. This results in a complete set of 38 dialogues. The number of conducted dialogues and user turns are given in table

1. In the first scenario, the user simply had to complete two tasks successively. Scenario two and three were selected to provoke situations where clarifications are required. In scenario two, a subdialogue should be executed, so that the user can interrupt a current dialogue, to fulfill a second task. After the second task, the first task should be completed. In scenario three, the user should start a dialogue with the system, which is then aborted to execute a second task. The problem for the system in scenario two and three is to differentiate if input, indicating a context switch, is intended by the user, or caused by speech recognition errors.

For evaluation purposes, we computed average completion rate and dialogue length from the log files. In addition, two subjective measures, the naturalness and the adequacy of dialogue length were given by the users on a feedback form. The naturalness was given on a scale from -2 to +2. The adequacy of dialogue length was given on a scale from -1 to 1, which means that the user was satisfied with the dialogue length (1) or not (-1). Table 2 shows the results of our small evaluation.

number of users	7
number of dialogues	38
number of utterances	353

Table 1: Overview of the data set.

The completion rate shows how many dialogue goals were finalized relative to the number of started goals. The dialogue length counts the number of turns between starting a dialogue goal and finalizing that goal.

	hold strategy (implicit)	anomaly analysis (explicit)
Objective measures from log files:		
completion rate	65%	81%
dialogue length	5	3
Subjective measures from user feedback:		
naturalness	-0.29	0.21
adequate length	0.79	0.79

Table 2: Comparison of strategies with implicit and explicit clarification requests.

In most categories the evaluated strategy is superior to the baseline system. In the given scenario, explicit clarification produces dialogues that are more robust than implicit clarification. This may be caused by the fact that the system directs the user's attention clearly to the current problem. It also shows that the user did not feel forced by the system to give redundant information, by assigning a mostly adequate length to both systems. Using explicit clarification leads to a higher completion rate and less user turns. Thus the naturalness of dialogues with explicit clarification is higher, which is also indicated by the feedback from the user.

8. Conclusions

In this paper we presented a strategy for explicit clarification requests. Through an anomaly analysis of dialogue state transitions we detect critical situations with need for clarification. In a finite state based clarification dialogue we solve the critical situation. Our approach is domain and language independent and easily expandable, by using explicitly defined triggers. A trigger is a rule set, which enables the anomaly analysis to detect

needs of clarification. In a small user evaluation we have shown that the use of explicitly asked clarification questions leads to a more clearly dialogue. Users are enabled to react specific to the actual system requirements.

9. Acknowledgements

Part of this work was funded by the German Research Agency (DFG) under Sonderforschungsbereich 588 - Humanoid Robots¹, and by the European Commission under project CHIL² (contract #506909).

10. References

- [1] M. Purver, J. Ginzburg, and P. Healey, "On the means for clarification in dialogue," in *Proceedings of the SIG-Dial 2001 Workshop on Discourse and Dialogue*, Aalborg, Denmark, 2001.
- [2] M. Gabsdil, "Clarification in spoken dialogue systems," in *Proceedings of the 2003 AAI Spring Symposium. Workshop on Natural Language Generation in Spoken and Written Dialogue*, Stanford, CA, 2003.
- [3] D. Schlangen, "Causes and strategies for requesting clarification in dialogue," in *5th SIGdial Workshop on Discourse and Dialogue*, Cambridge/MA, May 2004.
- [4] H. Holzappel and P. Gieselmann, "A way out of dead end situations in dialogue systems for human-robot interaction," in *Humanoids 2004*, Los Angeles, 2004.
- [5] E. Maier, "Clarification dialogues in VERBMO-BIL," in *Proc. Eurospeech '97*, Rhodes, Greece, 1997, pp. 1891–1894. [Online]. Available: cite-seer.ist.psu.edu/477468.html
- [6] M. Denecke, "Rapid prototyping for spoken dialogue systems," in *Proceedings of the 19th International Conference on Computational Linguistics*, Taiwan, 2002.
- [7] B. Carpenter, *The Logic of Typed Feature Structures*. Cambridge University Press, 1992.
- [8] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, and M. Westphal, "The karlsruhe-verbmobil speech recognition engine," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP-97*, Munich, Germany, 1997.
- [9] H. Soltau, F. Metze, C. Fuegen, and A. Waibel, "A one pass- decoder based on polymorphic linguistic context assignment," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop, ASRU-2001*, Madonna di Campiglio, Trento, Italy, December 2001.
- [10] C. Fügen, H. Holzappel, and A. Waibel, "Tight coupling of speech recognition and dialog management - dialog-context grammar weighting for speech recognition," in *Proceedings of the International Conference on Spoken Language Processing, ICSLP 2004*, 2004.
- [11] M. Gavalda, "Soup: A parser for real-world spontaneous speech," in *Proceedings of the 6th International Workshop on Parsing Technologies (IWPT-2000)*, 2000.

¹<http://www.sfb588.uni-karlsruhe.de>

²<http://chil.server.de/>