

Correction Grammars for Error Handling in a Speech Dialog System

Hirohiko Sagawa

Teruko Mitamura

Eric Nyberg

Language Technologies Institute, Carnegie Mellon University

Pittsburgh, PA 15213, U.S.A.

{hsagawa, teruko, ehn}@cs.cmu.edu

Abstract

Speech recognition errors are inevitable in a speech dialog system. This paper presents an error handling method based on correction grammars which recognize the correction utterances which follow a recognition error. Correction grammars are dynamically created from existing grammars and a set of correction templates. We also describe a prototype dialog system which incorporates this error handling method, and provide empirical evidence that this method can improve dialog success rate and reduce the number of dialog turns required for error recovery.

1 Introduction

In a dialog system, speech recognition errors are inevitable and often make smooth communication between a user and a system difficult. Figure 1 shows an example of a dialog between a user and a system which illustrates a system error. The system misrecognized “Tokyo” in the user’s utterance (U1) as “Kyoto” (S3). If the system correctly recognized the user’s utterance, the user could answer “yes” at U3 and the weather is reported (S6). However, in this case, the user must correct the error at U3 and the turns from S4 to U5 are required to recover from the error. The dialog system must recognize the user’s response to the system error (correction utterance). Otherwise, more turns (or a

complete restart) will be required to correct the error. Therefore, an error handling method which corrects a system error and returns to the normal dialog flow smoothly is an important requirement for practical dialog systems.

Recent work related to error handling in speech dialog systems has mainly focused on error detection. Walker et al. (2000), Bosch et al. (2001) and Kazemzadeh et al. (2003) extracted several parameters (e.g., acoustic, lexical and semantic) from a dialog corpus, and analyzed the differences between correction utterances and the other utterances in a dialog. They also tried to detect system errors by using these parameters as input to machine learning methods. However, the issue of error recovery is not addressed.

Danieli (1996) and LuperFoy & Duff (1996) proposed error detection methods based on plan matching. An error is detected when the intention or the parameter expressed in the user’s utterance is not consistent with the system’s assumptions and/or limitations. In these studies, the correction utterances are assumed to be recognized correctly.

Kitaoka et al. (2003) proposed a method to detect system errors based on the similarity of speech patterns and hypotheses overlapping in the recognition result. They also proposed a method to improve the recognition accuracy for correction utterances by selecting a speech recognition grammar according to the results of the error detection.

The previous studies assumed that the rules for speech recognition or natural language processing of correction utterances were prepared in advance (Danieli, 1996; LuperFoy & Duff, 1996). These rules are indispensable because the correction utterance often includes the information required to correct the error. The correction utterance depends on the dialog context, especially on the user’s utterances prior to the system error. Therefore it is difficult for the system designer to prepare these rules in advance when the dialog flow becomes complex. To solve this problem, a method that can automatically create the rules to interpret correction utterances is desirable.

In this paper, we will propose a method to dynamically create the rules to recognize correction utterances and repair recognition errors based on the dialog context. A prototype dialog system which incorporates the proposed method has been developed,

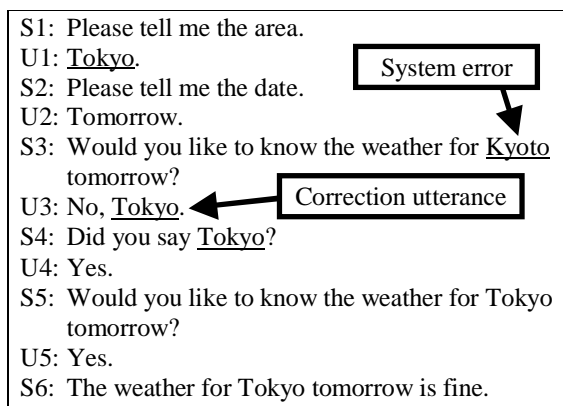


Figure 1. Example of a dialog with a system error

and we present experimental results which show the effectiveness of the approach.

2 CAMMIA Dialog System

Our current approach focuses on dialog systems which incorporate speech recognition modules utilizing regular grammars. The CAMMIA system is an example of such a dialog system (Nyberg et al., 2002).

The CAMMIA system is a client-server dialog management system based on VoiceXML. Each dialog scenario in this system is described in the format of DialogXML. The system has the initiative in the dialog, and dialogs are oriented around slot-filling for particular queries or requests. The server sends a VoiceXML data file to the client VoiceXML interpreter for a particular dialog turn, compiled from the DialogXML scenario according to the current dialog context. The VoiceXML data includes system prompts, names of grammar files and valid transitions to subsequent dialog states. The client interacts with the user according to the VoiceXML data.

Figure 2 shows an example of a grammar rule used in the CAMMIA system. The regular grammar rule can be represented as a transition network. The following sentences are recognized by the rule in Figure 2:

- I would like to know the weather for Tokyo.
- I would like to know the weather for Tokyo tomorrow.

3 Error Handling Based on Correction Grammars

To recognize the user's utterances in a dialog system, a grammar for potential user utterances must be prepared in advance for each dialog context. For error handling, it is also necessary to anticipate correction utterances and prepare a correction grammar. We propose a method to automatically create the correction grammar based on the current dialog context; error detection and repair is implemented using the correction grammar.

To create the correction grammar, the system must know the user's utterances prior to the error, because correction utterances typically depend on them. If the user's utterances are consistent with what the system is expecting, the correction grammar can be generated based on the grammar previously in use by the speech recognizer. Therefore, the sequence of grammars used in the dialog so far is stored in the grammar history as the dialog context, and the correction grammar is created using the grammars in this history.

Most of the forms of correction utterances can be expected in advance because correction utterances include many repetitions of words or phrases from previous turns (Kazemzadeh et al., 2003). We assume that the rules to generate the correction grammar can be prepared as templates; the correction grammar is created by inserting information extracted from the grammar

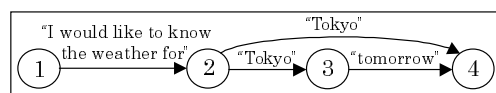


Figure 2. Example of the grammar rule used in the CAMMIA system

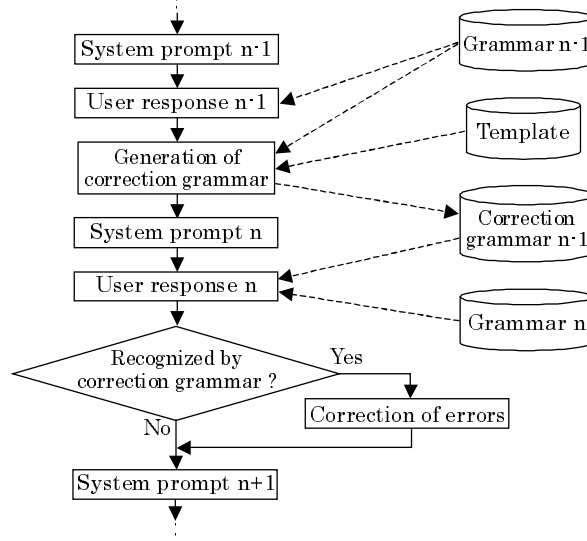


Figure 3. Process flow: Error handling based on a correction grammar

history into a template.

Figure 3 shows an example of a process flow in a dialog system which performs error handling based on a correction grammar. The "system prompt n" is the process to output the n-th prompt to the user. The correction grammar is created based on the grammar used in the "user response n-1", which is the process to recognize the (n-1)-th user utterance, and it is used in the "user response n" together with the "grammar n" which is used to recognize the n-th normal user's utterance. The system detects the error when the user's utterance is recognized using the correction grammar, and then transits into the "correction of errors" to modify the error. The grammar history in Figure 3 stores only the grammar used in the last recognition process. The number of grammars stored in the history can be changed depending on the dialog management strategy and error handling requirements.

4 Generation of Correction Grammar

The correction grammar is created as follows.

(1) Copying the grammar rules in the history

The user often repeats the same utterance when the system misunderstood what s/he spoke. To detect when the user repeats exactly the same utterance, the grammar rules in the grammar history are copied into the correction grammar.

(2) Inserting the rules in the history into the template

When the user tries to correct the system error, some

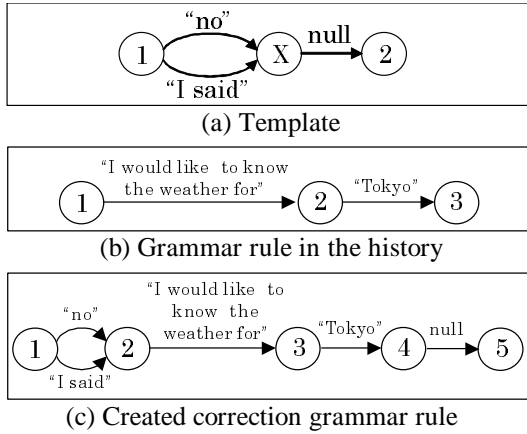


Figure 4. Correction grammar created by inserting the original rule into a template

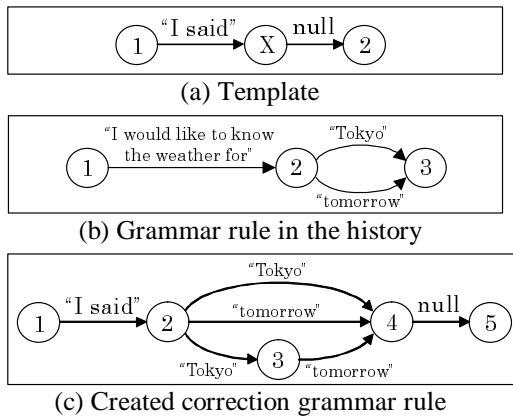


Figure 5. Correction grammar rules created by inserting slot values into a template

phrases are often added to the original utterance (Kitaoka, 2003). The template mentioned above is used to support this type of correction utterance. An example of the correction grammar rule generated by this method is shown in Figure 4. The “null” in Figure 4 implies a transition with no condition, and the “X” shows where the original rule is embedded. In this example, the created grammar rule in Figure 4(c) corresponds to the following sentences:

- No, I’d like to know the weather for Tokyo.
- I said I’d like to know the weather for Tokyo.

(3) Inserting slot-values into the template

The user often repeats only words or phrases which the system is focusing on (Kazemzadeh et al., 2003). In a slot-filling dialog, these correspond to the slot values. Therefore, correction grammar rules are also created by extracting the slot values from the grammar in the history and inserting them into the template. If there are several slot values that can be corrected at the same time, all of their possible combinations and permutations are also generated. An example is shown in Figure 5. In Figure 5(b), the slot-values are

“Tokyo” and “tomorrow”. The grammar rule in Figure 5(c) includes each slot value plus their combination(s), and represents the following sentences:

- I said Tokyo.
- I said tomorrow.
- I said Tokyo tomorrow.

5 Prototype System with Error Handling

We have implemented the proposed error handling method for a set of Japanese dialog scenarios in the CAMMIA system. We added to this system: a) a process to create a correction grammar file when the system sends a grammar file to the client, b) a process to repair errors based on the recognition result, and c) transitions to the repair action when the user’s utterance is recognized by the correction grammar.

There are two types of errors: task transition errors and slot value errors. If the error is a task transition error, the modification process cancels the current task and transits to the new task as specified by the correction utterance. When the error is a slot value error, the slot value is replaced by the value given in the correction utterance. However, if the new value is identical to the old one, we assume a recognition error and the second candidate in the recognition result is used. This technique requires a speech recognizer that can output N-best results; we used Julius for SAPI (Kyoto Univ., 2002) for this experiment.

6 Experiments

We carried out an experiment to verify whether the proposed method works properly in a dialog system. In this experiment, dialog systems with and without the error handling method were compared. In this experiment, a weather information dialog was selected as the task for the subjects and about 1200 dialog instances were analyzed (both with and without error handling). The dialog flow was the same as shown in Figure 1. The grammar included 500 words for place names, and 69 words for the date. The subjects were instructed in advance on the utterance patterns allowed by the system, and used only those patterns during the experiment. A summary of the collected data is shown in Table 1. When error handling is disabled, the system returns to the place name query when the user denies the system’s confirmation, e.g. it returns from U3 to S1 in Figure 1. A comparison of the number of turns in these two systems is shown in Table 2. “1 error” in Table 2 means that the dialog included one error and “2 errors” means that the same error was repeated.

The success rate for the task and the average number of turns in the dialog (including errors) are tabulated. Dialogs including more than 3 errors were regarded as incomplete tasks in the calculation of the success rate. The results are shown in Table 3.

Table 1. Summary of the collected data

	w/o error handling	w/ error handling
# of users	2 male, 1 female	2 male, 1 female
# of dialog	603	596
# of error dialog	66	61

Table 2. Number of turns in the dialog

	No error	1 error	2 errors
w/o error handling	7	13	19
w/ error handling		11	13

Table 3. Success rate and average number of turns

	Success rate	Ave. # turns
w/o error handling	86.4%	14.6
w/ error handling	93.4%	11.6

7 Discussion

The task completion rate was improved from 86.4% to 93.4% when the proposed error handling method was used. The average number of turns was reduced by 3 turns as shown in Table 3. This result shows that the proposed error handling method was working properly and effectively.

One reason that the success rate was improved is that the proposed method prevents the repetition of errors. When the error handling method is not implemented, errors can be easily repeated. The error handling method can avoid repeated errors by selecting the second candidate in the recognition result even when the correction utterance is also misrecognized. There were 7 dialogs in which the correction utterance was correctly recognized by selecting the second candidate.

However, there were 13 dialogs in which the error was not repaired by one correction utterance. There are two explanations. One is that there are insertion errors in speech recognition which causes words not spoken to appear in the recognition result. For example, the system prompt S4 for U3 in Figure 1 becomes as follows:

S4: Did you say Tokyo yesterday?

In this case, the user has to speak more correction utterances. The second explanation is that the recognition result did not always include the correct result within the first two candidates. It is not clear that extending the repair mechanism to always consider additional recognition candidates (3rd, 4th, etc.) is a viable technique, given the drop off in recognition accuracy; more study is required.

8 Conclusions

In this paper, we proposed an error handling method based on dynamic generation of correction grammars to recognize the user corrections which follow system errors. The correction grammars detect system errors and also repair the dialog flow, improving task

completion rates and reducing the average number of dialog turns. We developed a prototype dialog system using the proposed method, and demonstrated empirically that the success rate improved by 7.0%, and the number of turns was reduced by 3.

The creation of rules for correction utterances based on the dialog history could be applicable to dialog systems which use speech recognition or natural language processing and other kinds of rules beyond regular grammars; we plan to study this in future work. We are also planning to develop an algorithm to improve the precision of corrections that are based on the set of recognition candidates for the correction utterance and an error recovery strategy. We also plan to apply the proposed method to other types of dialogs, such as user-initiative dialogs and mixed-initiative dialogs.

References

- Abe Kazemzadeh, Sungbok Lee and Shrikanth Narayanan. 2003. *Acoustic Correlates of User Response to Error in Human-Computer Dialogues*, Proceedings of ASRU 2003: 215-220.
- Antal van den Bosch, Emiel Kraemer and Marc Swerts. 2001. *Detecting problematic turns in human-machine interactions: Rule-Induction Versus Memory-Based Learning Approaches*, Proceedings of ACL 2001: 499-507.
- Eric Nyberg, Teruko Mitamura, Paul Placeway, Michael Duggan, Nobuo Hataoka. 2002. *Dynamic Dialog Management with VoiceXML*, Proceedings of HLT-2002.
- Kyoto University, 2002, Julius Open-Source Real-Time Large Vocabulary Speech Recognition Engine, <http://julius.sourceforge.jp>.
- Marilyn Walker, Jerry Wright and Irene Langkilde. 2000. *Using Natural Language Processing and Discourse Features to Identify Understanding Errors in a Spoken Dialogue System*, Proceedings of ICML-2000: 1111-1118.
- Morena Danieli. 1996. *On the Use of Expectations for Detecting and Repairing Human-Machine Miscommunication*, Proceedings of AAAI Workshop on Detection, Repair and Prevention of Human-Machine Miscommunication: 87-93.
- Norihide Kitaoka, Kaoko Kakutani and Seiichi Nakagawa. 2003. *Detection and Recognition of Correction Utterance in Spontaneously Spoken Dialog*, Proceedings of EUROSPEECH 2003: 625-628.
- Susann LuperFoy and David Duff. 1996. *Disco: A Four-Step Dialog Recovery Program*, The Proceedings of the AAAI Workshop on Detection, Repair and Prevention of Human-Machine Miscommunication: 73-76.