A Comparison of Confirmation Styles for Error Handling in a Speech Dialog System

Hirohiko Sagawa, Teruko Mitamura, Eric Nyberg

Language Technologies Institute
Carnegie Mellon University, USA
{hsagawa, teruko, ehn}@cs.cmu.edu

Abstract
Speech recognition errors are inevitable in a speech dialog system. It is important to provide a dialog flow that allows the user to correct system errors and quickly return to the normal dialog flow smoothly. Therefore, an error handling method which allows a user to recover from the error. If the system does not support such a confirmation procedure, the usability of the system will decline. Therefore, an error handling method which allows a user to correct an error and return to the normal dialog flow smoothly is an important requirement for practical dialog systems.

We have proposed an error handling method to detect correction utterances which follow system errors using dynamically generated correction grammars [1]. However, the dialog flow that takes place when a system error occurs was not addressed. In an erroneous situation in a dialog, a confirmation procedure is important for a user to identify a system error and modify it smoothly [2][3]. Possible confirmation styles can be classified into two types: explicit and implicit confirmations [4]. Explicit confirmation, in which the user is asked directly for each utterance, is most reliable, but often increases the number of turns [5][6]. Implicit confirmation, in which the next query from the system includes what the user said, has an advantage in dialog time and speed, but may confuse the user when the system prompt includes an error [3][4]. Recent work related to the confirmation procedure has mainly focused on how to reduce the number of turns in the explicit confirmation style by detecting erroneous recognition of the user’s utterance.

Komatani et al. [5] proposed a method to control the confirmation subdialog based on the confidence score in the speech recognition result. Smith [7] used a confidence score based on natural language processing. In these methods, the system confirms the user’s utterance using explicit confirmation when the confidence score is below a threshold value determined in advance.

Sturm et al. [8] proposed a method to use explicit confirmation when the confidence score based on the speech recognition result is below a threshold; otherwise implicit confirmation is used. The method by Lamel et al. [9] selects explicit confirmation when new information from the user is contradictory with information that was previously received; otherwise implicit confirmation is selected.

These methods are intended to improve dialog efficiency based on the number of turns, the dialog success rate and dialog time. The evaluation results showed that they could improve dialog efficiency when compared with a system only using explicit or implicit confirmation. However, an evaluation from the perspective of system usability was not conducted. Both usability and dialog efficiency must be acceptable in a practical dialog system. Also, previous studies have not considered an evaluation of the dialog flow when a system error is encountered. For an error handling method, we must determine the optimal combination of confirmation styles by evaluating differences in usability, both in normal dialogs and in dialogs containing system errors.

In this paper, three types of confirmation styles were evaluated from a viewpoint of usability to determine an optimal dialog flow for error handling. The combination of confirmation styles in normal dialogs and in dialogs with system errors will be discussed in the context of the evaluation results.

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). 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. If the system does not support such a dialog flow, the usability of the dialog system will decline. Therefore, an error handling method which allows a user to correct an error and return to the normal dialog flow smoothly is an important requirement for practical dialog systems.

We have proposed an error handling method to detect correction utterances which follow system errors using dynamically generated correction grammars [1]. However, the dialog flow that takes place when a system error occurs was not addressed. In an erroneous situation in a dialog, a confirmation procedure is important for a user to identify a system error and modify it smoothly [2][3]. Possible confirmation styles can be classified into two types: explicit and implicit confirmations [4]. Explicit confirmation, in which the user is asked directly for each utterance, is most reliable, but often increases the number of turns [5][6]. Implicit confirmation, in which the next query from the system includes what the user said, has an advantage in dialog time and speed, but may confuse the user when the system prompt includes an error [3][4]. Recent work related to the confirmation procedure has mainly focused on how to reduce the number of turns in the explicit confirmation style by detecting erroneous recognition of the user’s utterance.

Komatani et al. [5] proposed a method to control the confirmation subdialog based on the confidence score in the speech recognition result. Smith [7] used a confidence score based on natural language processing. In these methods, the system confirms the user’s utterance using explicit confirmation when the confidence score is below a threshold value determined in advance.

Sturm et al. [8] proposed a method to use explicit confirmation when the confidence score based on the speech recognition result is below a threshold; otherwise implicit confirmation is used. The method by Lamel et al. [9] selects explicit confirmation when new information from the user is contradictory with information that was previously received; otherwise implicit confirmation is selected.

These methods are intended to improve dialog efficiency based on the number of turns, the dialog success rate and dialog time. The evaluation results showed that they could improve dialog efficiency when compared with a system only using explicit or implicit confirmation. However, an evaluation from the perspective of system usability was not conducted. Both usability and dialog efficiency must be acceptable in a practical dialog system. Also, previous studies have not considered an evaluation of the dialog flow when a system error is encountered. For an error handling method, we must determine the optimal combination of confirmation styles by evaluating differences in usability, both in normal dialogs and in dialogs containing system errors.

In this paper, three types of confirmation styles were evaluated from a viewpoint of usability to determine an optimal dialog flow for error handling. The combination of confirmation styles in normal dialogs and in dialogs with system errors will be discussed in the context of the evaluation results.

2. System overview
This section presents an overview of the speech dialog system that formed the basis for our experiments on error handling.

2.1. CAMMIA dialog system
Our current approach focuses on dialog systems which incorporate speech recognition modules utilizing regular grammars. The CAMMIA system has been developed as such a dialog system [10].

CAMMIA is a client-server dialog management system based on VoiceXML. Each dialog scenario 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. Plural tasks are supported by the system, and the user can freely switch between tasks, even when the current task is not complete. The server sends a VoiceXML data file to the client Voicexml interpreter for a particular dialog turn; the VoiceXML is 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.

### 2.2. Error handling method

We previously proposed an error handling method based on dynamic generation of correction grammars to recognize the correction utterances which follow system errors [1]. In this method, the sequence of grammars used in the dialog so far is stored in the grammar history, and the correction grammar is created using the grammars in this history. The correction utterance is detected using the created correction grammar, and the system error is corrected based on the recognition result for the correction utterance.

Figure 2 shows 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 “generation of correction grammar” creates the correction grammar based on the grammar used in the “user response n-1” which is the process to recognize the (n-1)-th user utterance. The grammar history in Figure 2 is supposed to store only the grammar used in the last recognition process. The created correction grammar 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 number of grammars stored in the history can be changed depending on the dialog management strategy and error handling requirements.

The correction grammar is created by following three algorithms:

- Copying the grammar rules in the history;
- Inserting the rules relevant to slot-values in the history into the template.
- Inserting the rules relevant to slot-values in the history into the template.

The templates are possible patterns of the correction utterances. We determined these patterns by referring to previous studies such as [11] which analyzed the correction utterances in speech dialog systems.

In the “correction of errors”, the current task is canceled and the new task specified by the correction utterance is initiated when the user tries to correct an erroneous task transition that occurred due to a recognition error. When the correction utterance modifies the slot value, the old 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 [12].

### 3. Experiment design and method

#### 3.1. Dialog flows

A weather information dialog in Japanese was selected as the subject task. The weather information dialog includes two slots (area and date). The system asks the user for the area name and the date, one by one, and outputs a weather report. The following confirmation styles were selected for this experiment:

1. explicit confirmation
2. final confirmation
3. implicit confirmation

Dialog systems using these confirmation styles were constructed. Dialog examples for each confirmation style are shown in Figure 3, Figure 4 and Figure 5 respectively. Each example includes one system error and one correction utterance. SE02 in Figure 3, SF03 in Figure 4 and SI02 in Figure 5 are the system prompts to confirm both the slot values at the same time. In implicit confirmation, the system speaks the next query or the weather report after repeating what the user said (SI02, SI03 and SI04 in Figure 5).

---

**Figure 2**: Process flow: Error handling based on a correction grammar

**Figure 3**: Dialog example with explicit confirmation

| SE01: 地域を教えて下さい。(Please me the area.) | UE01: 東京の天気ですね。 (The weather for Tokyo.) |
| SE02: はい、京都ですがですか。 (Uh-huh. Did you say Kyoto?) | UE02: いいえ、東京です。 (No, Tokyo.) |
| SE03: はい、東京です。 (Uh-huh. Did you say Tokyo?) | UE03: はい、 (Yes.) |
| SE04: それは明日の天気を教えて下さい。 (Then please tell me the date.) | UE04: 明日の天気ですね。 (Tomorrow.) |
| SE05: はい、明日でよろしいですか。 (Uh-huh. Did you say tomorrow?) | UE05: はい、 (Yes.) |
| SE06: 東京は明日、晴れでしょう。 (The weather for Tokyo tomorrow will be fine.) | UE06: 明日、晴れでしょう。 (The weather for Tokyo tomorrow will be fine.) |
Subjects were asked to inquire about the weather information for 20 combinations of area and date. They repeated such inquiries for each dialog system described above. When the subjects got only a few system errors, they were asked to repeat the dialogs. To cancel the effect of the order in which the subject used the systems, the order was changed for every subject. The subjects were taught about possible dialog flows and expressions (normal utterances and correction utterances) before they tried each system. All of the system prompts and the user’s utterances were logged during the experiment.

The possible correction utterances were created by the method described in 2.2. They were the repetitions of the previous sentences or the phrases corresponding to the slot values. The subject could also add several words before or after these sentences or the phrases corresponding to the slot values. The subjects were taught about possible dialog flows and expressions (normal utterances and correction utterances) before they tried each system. Examples of a system prompt and subsequent correction utterances are shown below:

**[System prompt]**
京都の明日の天気でよろしいですか。(Would you like to know the weather for Kyoto tomorrow?)

**[Correction utterances]**
(a) 東京の天気を教えて下さい。(Please tell me the weather for Tokyo.)
(b) はい、東京です。(Yes.)
(c) いいえ、東京の天気です。(No, the weather for Tokyo.)
(d) 東京の明日の天気です。(The weather for Tokyo tomorrow will be fine.)

The subjects were also taught to always try to correct the system error using a correction utterance, and to continue until each dialog was completed, even if many errors repeated.

The speech recognition module was intentionally configured to randomly generate errors, since the purpose of the experiment was to evaluate usability in erroneous situations. The recognition accuracy for the slot values was set to about 70% on the average.

After trying all three systems, the subjects answered a questionnaire in which they ranked the systems from a viewpoint of usability. The items in the questionnaire were as follows:

1. The dialog flow is natural
2. The system is easy to use
3. The user wants to use it

For each item, three types of confirmation styles were compared in the case where there was no error and in the case where the errors were included, respectively. The user assigned each confirmation style the number, 1, 2 or 3, where 1 means the best and 3 the worst.

### 3.3. Result
13 subjects (4 adult Japanese males and 9 adult Japanese females) participated in this experiment. Two of them had prior experience using similar dialog systems; the others were native users. They used only the expressions taught in advance of the experiment. A summary of the dialogs logged during the experiment is shown in Table 1. The average accuracy of speech recognition was in reality a little lower than the 70% setting. The average ranks for each system in the cases where there was no error and where errors were included are shown in Table 2 and Table 3 respectively.

A two-way repeated-measures ANOVA and a set of t-tests were performed for the average ranks. The ANOVA shows that there is a significant difference between the cases where there was no error and where the error was included (p<0.05). And the t-test shows that there are significant differences between the explicit and implicit confirmations with no error (p<0.05) and between the explicit and implicit confirmations.
with errors (p<0.05).

A two-way repeated-measures ANOVA and a set of t-tests were also applied to the ranks for each item in the questionnaire. There were significant differences between ranks for (1) natural dialog flow and between ranks for (3) want to use (p<0.05). The significant differences between the explicit and final confirmations (p<0.05) and between the explicit and implicit confirmations (p<0.05) with no error were shown for (1) natural dialog flow. The significant difference between the explicit and implicit confirmations (p<0.05) with errors was shown for (3) want to use.

4. Discussion

From the result of the ANOVA, the evaluation for the confirmation style when there was no system error was clearly different from that when system errors were included. In Table 2, the rank for the final confirmation is highest and for the explicit confirmation lowest. On the other hand, the explicit confirmation has the highest rank and the implicit confirmation the lowest rank in Table 3. The t-test showed that the differences between two types of confirmation styles in each case were significant. These results mean that the user clearly prefers the style with few confirmation turns when there is no error, and the style to surely correct errors when the error occurs. The previous studies proposed the methods to decrease the number of explicit confirmation turns to improve the dialog efficiency [5][7][8][9]. The results in this experiment assure that these methods are also appropriate in a dialog system from the viewpoint of usability.

There was no significant difference between the final and implicit confirmations when no error was included in the dialog. However, in Table 2, there is a tendency that the user prefers the final confirmation rather than the implicit confirmation even though the number of turns in the final confirmation is greater than that in the implicit confirmation. Some of the subjects told us that they were sometimes confused with the final confirmation because it is often used in the daily life. We guess this is the explanation for this result.

When errors were included in the dialog, the rank for the final confirmation was lower than the explicit confirmation. It seems that confirming each slot value one by one is preferred rather than confirming all of them at the same time by the user, though the t-test did not show a significant difference between these two styles. In the final confirmation, there were four cases where both of the slot values were correct, only the first one was correct, only the second one was correct and both of them were incorrect when they were confirmed. Some of the subjects told us that they were sometimes confused when the error was included. We guess they preferred the explicit confirmation more than the final confirmation because the error in the explicit confirmation could be identified easier than in the final confirmation.

From these results, we found that the combination of the final and explicit confirmation styles provides the best dialog management from the viewpoint of usability, although there is concern about potential user confusion with final confirmation. Our error handling method is based on the user’s correction utterance. The combination of two confirmation styles can be achieved by using final confirmation in a normal dialog, and transiting to a dialog flow using explicit confirmation when a correction utterance is detected.

5. Conclusions

In this paper, we presented three types of confirmation (explicit, final and implicit) and compared them from the viewpoint of usability. From the results, it became clear that users prefer final confirmation in a normal dialog, and explicit confirmation in a dialog with errors.

Currently, we are implementing this dialog flow in our system. We will also study how to allow the user to easily identify system errors when several slot values are confirmed at the same time. In this experiment, the number of slots in the dialog was limited to two, and the recognition accuracy was set relatively low. The evaluation for these confirmation styles might change when the dialog becomes more complex and recognition accuracy is improved. We need to make more comparisons of confirmation styles by changing the number of slots and accuracy of speech recognition.

6. References