# The Impact of Response Wording in Error Correction Subdialogs

*Julie Goldberg*†, *Mari Ostendorf*‡, *Katrin Kirchhoff*‡

†Computer Science & Engineering, ‡Electrical Engineering Departments
University of Washington, Seattle, WA USA
julie@cs.washington.edu,{mo,katrin}@ee.washington.edu

## Abstract

Spoken human-machine dialogs are prone to communication failures due to imperfect speech recognition and understanding. In order to recover from these failures, users typically engage in error correction subdialogs. Lengthy error correction subdialogs should be avoided since they increase the overall task completion time and decrease user satisfaction. This study analyzes a large corpus of human-computer dialogs and identifies properties of system responses that affect user frustration and recognition error rates in error correction subdialogs.

## 1. Introduction

When users realize that a recognition or understanding error has occurred in a spoken dialog system, they typically initiate a subdialog aimed at recovering from the misunderstanding. An example of these so-called error correction subdialogs is shown below (SYS = system prompt, USR = user utterance).

SYS1: Let's see then, I have you going from Newark to Dhaka on October twenty-fifth. Is that correct?
USR1: no
SYS2: Oops, let's try again. I have you arriving in Dhaka. Have I got that right?
USR2: no
SYS3: What is your destination?
USR3: the Bahamas
SYS4: What is your destination?
USR4: the Bahamas
(repeated 3x)
SYS8: What city in Nevada would you like to go to?
USR8: no
SYS9: Oops, let's try again. I have you arriving in Dhaka. Have I got that right?

Error corrections can lead to user frustration if not properly handled. Error handling involves (a) detecting that an error has occurred or that the user wants to correct (or change) an entry, (b) changing the dialog manager to an error correction mode, and (c) generating spoken responses that are appropriate to that situation. In this study we look at the third problem – response generation – in the context of a mixed-initiative dialog system. The focus here is not on dialog strategy, as in [1], but rather on the wording of the responses. While response generation is intimately linked to the dialog strategy, some variations in phrasing apply to both open and constrained system queries.

Prior work has shown that users of dialog systems often change their wording (or their modality in multi-modal systems) after discovering a system error [2, 3, 4]. This raises the question of whether rephrasing is also a good strategy for designing system prompts, and, in particular, whether it is helpful for response generation in error correction subdialogs. In addition, we raise the question of whether system apologies are useful in the context of a suspected error.

To answer these questions, we conducted a study based on a large corpus of human-computer interactions from the NIST 2000 Communicator Evaluation [5], which includes data from 9 different mixed-initiative telephone-based dialog systems. In particular, we looked at different user behaviors (hyperarticulation, frustration, and rephrasing) and analyzed recognition system error rates as a function of different types of system responses (apologies, rephrasing, etc.). Our analysis focused on system "repromptings," where the system asked for the same information multiple times in a row. The primary goal is to determine whether the choice of system response can help direct the user to produce utterances that are easier to recognize, resulting in shorter error correction subdialogs. A secondary goal is to characterize system responses least likely to further frustrate users in these already-irritating situations. In the sections to follow, we describe prior work that motivates some of the questions posed here as well as the factors controlled for, followed by a description of the analysis method and experimental results. The findings – that apologies and rephrasing are useful – have important implications for response generation that are relatively straightforward to implement.

## 2. Background

Studies related to dialog systems design have looked at human-human dialogs (e.g. [6]), Wizard-of-Oz experiments (e.g. [6, 3, 2]), and genuine human-machine interactions (e.g. [7, 4]). Previous work on error handling in either Wizard-of-Oz or human-machine interaction scenarios has focused on linguistic and phonetic properties of corrections, the impact of the overall dialog strategy on corrections, and on the automatic detection of error corrections or easily misrecognizable user utterances [8, 9, 10, 11, 12, 13]. However, there has been little work relating error subdialogs to properties of response generation, which is the main objective of this study.

Oviatt et al. [2, 3] studied users' responses to system errors within a Wizard-of-Oz form-filling task allowing multimodal input (either speech or handwriting). Recognition errors were simulated by randomly asking the user to retry their input rather than displaying the information just provided. An analysis of how users' utterances changed during sequential repetitions showed that users often use an exact repetition once and then change their strategy, either switching modalities (from speech to handwriting or vice versa) or changing the lexical content of their utterances.

Swerts et al. [4] investigated the content change of user utterances after errors due to misrecognitions and the content of user utterances after system rejections. Their study is based on the TOOT train scheduling task. User responses were categorized into five classes: Add, Add/Omit, Omit, Paraphrase and Repeat, depending on how an utterance compares to the prior one it reiterates. It was found that omitting information was the most common strategy when correcting a misrecognition error, followed by repetitions and paraphrases. Both omitting and adding information were rare after rejection errors; users usually repeated or paraphrased their utterances. The focus of this study was on the influence of dialog strategy on users' error-correction strategies; the influence of the users' strategy on word error rate was not investigated.

Shin et al. [1] compared how users discover errors based on system behavior (explicit confirmation, implicit confirmation, help, system repeat, reject, non sequitur). The study is based on 161 dialogs from the 664-dialog NIST Communicator corpus [5]. It was found that users take longer to get back on track (and fail more often) when they discover errors through implicit confirmations and non sequiturs rather than through explicit confirmations.

Other studies have also investigated the impact of dialog strategy and input modality (handwriting, speech, etc.) on user inputs and user satisfaction [14, 15, 7]. However, they did not focus on error situations.

Levow [8], Oviatt et.al [16] and Swerts et al. [4] also studied the phonetics of users' corrections, which can help with automatic error detection. It was generally found that error corrections have acoustic and prosodic properties that differ from those of normal user utterances. Error corrections are distinguished by increased average duration and wider F0 and energy ranges. Error corrections had a higher recognition error rate on average, and most of the error corrections were audibly hyperarticulated. Speech recognition errors were also increased with increasing depth into the error correction subdialog [4].

There has been little work on how a system prompt influences the subsequent user utterance when the system uses only speech input/output, but error conditions highlight the importance of this problem. Our ultimate goal is to change a system's prompts to facilitate recognition and understanding of the subsequent user utterance. This should shorten error subdialogs and increase user satisfaction.

# 3. Method

In human-computer dialogs, the system often asks for the same information multiple times in a row. This happens whenever the system does not receive the information it requested, either because the user did not answer the system's question (rare), or because the system made a recognition or understanding error, as in the SYS4 and SYS8 responses in the introductory example. We refer to such situations as *repromptings*. We investigated how reprompting in dialog systems influences users and how the different user responses influence recognizer performance.

## 3.1. Corpus: NIST 2000 Evaluation Data

We analyzed the transcripts of conversations from the NIST 2000 Communicator Evaluation [5]. In this corpus, 72 paid users called and attempted to make travel arrangements using 9 different mixed-initiative dialog systems. Users were asked to arrange 9 different travel scenarios (7 specified, 2 open), calling each of the systems once. Not all users completed all 9 calls, so there are 664 dialogs in the corpus. Repromptings were found in 521 (78%) of the dialogs; out of 12004 user utterances, 2733 (23%) are responses to system repromptings.

Many of the dialogs have been hand-labeled for various properties. As part of this study, 149 (27%) of the dialogs, containing 2799 (23%) utterances were hand-labeled for hyperarticulation by a native speaker of English (and spot checked by a second labeler). Another subset of the corpus was annotated (by ICSI and SRI) with emotion labels [17], including 392 (59%) of the dialogs, containing 7546 (63%) of all utterances. Many dialogs were not labeled, because they were very short or because there was only one sound file for an entire dialog (rather than one per utterance). Of the 149 dialogs labeled for hyperarticulation, 80 were also labeled for emotion. Our analysis with these labels compared two emotion categories: *angry/frustrated* grouped with *annoyed* (henceforth just "frustrated") versus all other categories (*neutral, tired, amused* and *other*).

## 3.2. System Response Classifications

We extracted dialog segments (two successive pairs of system prompts and user utterances) where the system is prompting for the same information twice in a row. We classified the segments according to three different criteria: *Spiral Depth*, *Desired Answer* and *Reprompt Manner*.

*Spiral Depth* and spiral errors are terms introduced by Oviatt [2]. A spiral error is a sequence of repeated misunderstandings within the same error correction subdialog. During spiral errors, users repeat their responses over and over, as the system repeatedly misrecognizes their utterances. Spiral depth describes how deep into a spiral error the user has gone. The first user repetition is classified as spiral depth one; the second as spiral depth two; etc. As described above, studies have shown that user behavior and ASR error change with spiral depth; hence, we include it as a condition of our analyses.

*Desired Answer* characterizes the information the user is being asked to provide. These categories are used for normalization when comparing word error rates, because system performance varies across categories. For example, most systems have a higher error rate on cities, but systems vary in how well they recognize "yes" and "no". System queries are classified into "Airline", "City", "Date", "Time", "Yes/No", "Correct" and "Other". (The responses to "is that correct" pattern differently than those to other yes/no questions, since they often involve corrections. Hence, these responses are split into two categories.)

*Reprompt Manner* is the focus of this study. When a system has to ask users to give the same information they just gave, it can repeat the query exactly, or it can change the prompt in many different ways. We classify segments into one of seven different Reprompt Manner categories:

- **Repeat** Exact repetition of previous utterance.
- **Rephrase** The prompt is asked in a different way the second time, sometimes with instructions to the user.
- **Partial Repeat** The prompt is an exact substring of the prior prompt, or the query portion of both are identical.
- **Not Understood** The prompt is along the lines of "I'm sorry. I didn't understand." or "Pardon me?" These reprompt without asking for any specific content.
- **Apology Repeat** Same as Repeat, but the second prompt begins with an apology.
- **Apology Rephrase** Same as Rephrase, but the second prompt begins with an apology.
- **Apology Partial Repeat** Same as Partial Repeat, but the second prompt begins with an apology.

The distinction between "Partial Repeat" and "Rephrase" is somewhat blurred, since many partial repeats can also be considered rephrasings. For purposes of this study, an utterance is labeled a Partial Repeat if one string is a substring of the other, excluding confirmations. Under this definition, we label the second utterance in

> SYS1: Okay, from West Palm Beach to Salt Lake City on Monday October 2. Can you provide the approximate departure time or airline?
> SYS2: Okay, can you provide the approximate departure time or airline?

as a partial repeat, and

> SYS1: Flying from Honolulu to Chicago O'Hare, what date would you like to fly?
> SYS2: Flying from Honolulu to Chicago O'Hare, please tell me what date you wanna travel.

as a rephrase.

Classifying all the system reprompts was done semi-automatically, without access to the internal state of any system but simplified by the fact that all the systems used template-based language generation. Exact system repeats included all those that differed only in punctuation, the word "and" or the word "please". Rephrasings were found using an iterative process of checking unclassified prompts in a subset of the data and adding key phrases to a list of known types of rephrasing. Apologies and Not Understood prompts were found by searching for key phrases containing "sorry", "pardon", and other cue words determined by inspection of the data. Within spiral errors, we try to match the new prompt with the last prompt asking for content (i.e. skipping Not Understood prompts). If it matches, we label the segment with the corresponding reprompt manner. Otherwise, it is not a reprompt. Prompts were categorized by desired answer according to keywords such as "destination" and "city". Prompts that did not fit neatly into one category (e.g. "Okay, can you provide the approximate departure time or airline?", "What are your travel plans") were place into a separate category *Other*.

Because of the automation, some data is missed or misclassified. The better the recall, the more data can be analyzed. For this study, noisy data is worse than less data, so we focused on precision rather than recall. To check the accuracy of our classification procedure we hand-labeled 36 dialogs, 4 from each system, with a total of 128 system repromptings. Three of the dialogs did not contain any repromptings in them, but we had at least 3 dialogs with repromptings from every system. Overall, recall was 92%, precision was 97% (or 98% if partial repeat and rephrase are merged) and accuracy on spiral depth was 97%. The spiral depth errors come from a single reprompting in a spiral error that was not identified as such; three subsequent repromptings had incorrect spiral depth. Since the desired answer categories are used only for normalization purposes, our objective was high precision in the non-other categories (100% in the hand-checked set) rather than recall (not measured).

### 3.3. Measurements

In analyzing user behavior, we measured the rates (percentage) of frustration, hyperarticulation and user repeats under different conditions. We also looked at two measures of recognizer performance, described below, as indicators of the potential to correct errors and end an error subdialog.

**Word Error Rate (WER)** indicates how well the user's utterance was recognized by the system. WER was computed using the standard NIST scoring software *Sclite* (`http://www.nist.gov/speech/tools`). Word error rates reported here ignore disfluencies, including filled pauses, fragments and simple repetitions. We used WER measures directly in assessing the impact of different user behaviors on recognizer performance.

The different recognizers have considerable performance differences, with averages ranging from 24-41%. Within systems, there are also general patterns of which desired answer categories were easier or harder for systems to recognize. For example, error rates on cities (open class) were higher than the overall error rate, while error rates on dates were lower. Further, the degree of difference varied greatly from system to system. Therefore, word error rates cannot be compared directly for response types, so we normalized for system and answer category variation by using a ratio of word error rates. The ratio for a particular response type X, desired answer DA and system S is:

$$R_e(S, DA, X) = \frac{WER(S, DA, X)}{WER(S, DA)}$$

The type $X$ could be all utterances after one reprompt manner, at one spiral depth, or any other grouping. When there are $< 40$ sentences or $< 80$ words, we back off to the system-level WER, i.e. $WER(S, DA) \approx WER(S)$. Weighted averages can be computed across systems to given an average **WER ratio:**

$$R_e(X) = \frac{1}{n(X)} \sum_S \sum_{DA} n(S, DA, X) R_e(S, DA, X)$$

where $n(\cdot)$ is the total number of words in utterances in the specified class. This measure allows for system-independent comparisons of recognizer performance as a function of variable $X$.

### 3.4. Statistical Significance Tests

For binary distinctions (e.g. frustration, hyperarticulation), we used the standard pairwise t-test to determine significance. WER roughly follows a binomial distribution, so the t-test is used here as well. Because the sample sizes are large, the Gaussian model used in the t-test is a reasonable approximation. The t-test is not a good choice for the cross-system WER ratio comparisons. We instead use the standard non-parametric Wilcoxon Rank Sum test to determine if the two WER ratios differ significantly. Each word is a sample, with the (1/0) error value scaled by the normalization factor $WER(S, DA)$. In this paper, when we say a difference is significant, it is significant at $p \leq 0.01$.

## 4. Results

In our analyses, described below, we look at how different user behaviors affect recognition performance, how the system's response influences the user's behavior, and whether the system response's observed impact on behavior translates directly into a change in WER.

### 4.1. Impact of User Behaviors on Word Error Rate

It has been observed that a user's speaking style impacts word error rate [18]. For the problem of improving error correction subdialogs, we look at the impact of frustration and hyperarticulation. In addition, during reprompts, we look at the impact of user repetition vs. rephrasing.

User hyperarticulation did not have the negative effect on recognizer performance that we expected, as shown in Fig. 1.

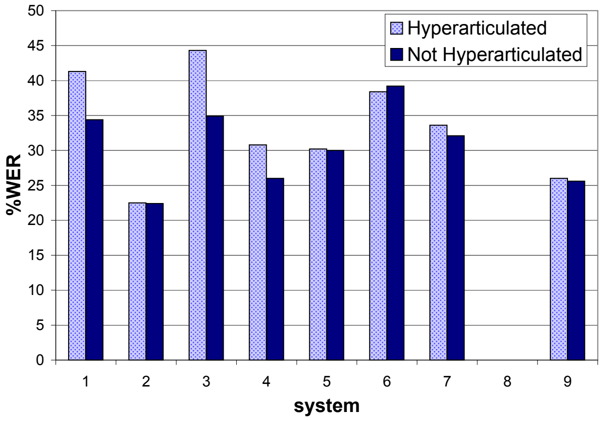

Figure 1: Hyperarticulation Effects on Word Error Rate

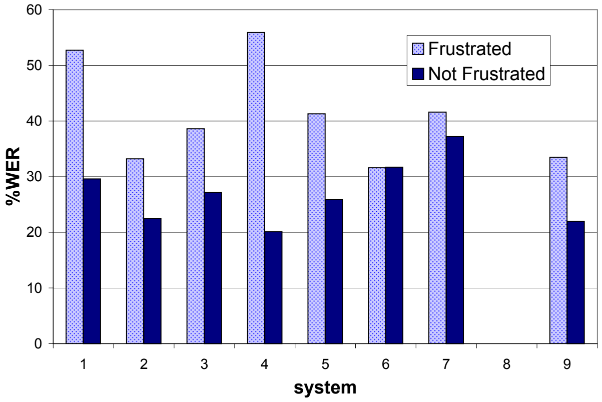

Figure 2: WER of different systems as a function of frustration labels.

It only significantly hurts the performance of system 3, though it probably effects the performance of systems 1 and 4 as well (p = 0.04, p = 0.11 respectively). Hyperarticulation had no significant effect on the other systems. To our knowledge, none of these systems use a special strategy to deal with hyperarticulated speech; however, there probably were differences in the training data used to build the different recognizers. Since hyperarticulation did not seem to be an important factor for all systems, we did no further analysis with it.

An important goal in dialog system design is to avoid user frustration. This is a valid goal in its own right, but as Fig. 2 shows, frustrated users also speak in a way that is harder for most systems to recognize. Every system except 6 and possibly 7 (p = 0.12) performed significantly worse on frustrated utterances. Ang et al. [17] found that hyperarticulation and frustration are not highly correlated, so this does not contradict the above result.

Our focus is on reprompts, not just general user utterances. In reprompts, there are two user utterances as well as two system prompts, and since the first utterance has already failed to be recognized, the goal is to recognize the second one better. As is shown in Fig. 3, recognizers performed significantly better when users rephrased their utterance rather than repeating it word for word. This could be due to either (or both) a change in speaking style when there is no change in wording or the fact that the original utterance has out-of-vocabulary (OOV) words. We do not have access to OOV information, but we did observe

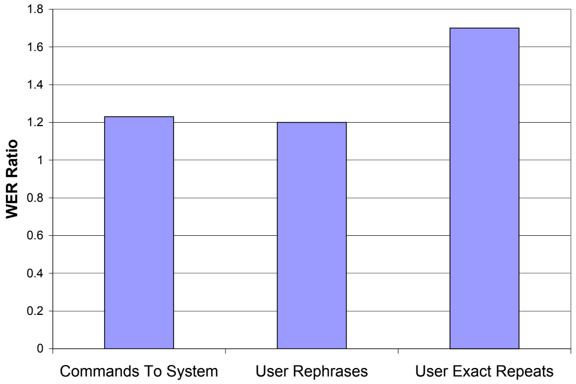

Figure 3: Normalized WER as a function of the type of user response to a reprompt.

that repeats have significantly higher rates of both frustration and hyperarticulation than rephrases (34% vs. 24% frustrated, and 68% vs. 50% hyperarticulated). ASR also performed significantly better on commands to the system than on repeats, but those are limited to “Back up”, “Start over” and “Scratch that.” There is no significant difference between system commands and rephrases, however, so there is no reason to encourage system commands, and there are dialog strategy reasons to avoid it – starting from scratch should be a last resort.

Because it helps so much when the user rephrases their utterance, we wanted to measure the extent of rephrasing. We tried a word-level distance metric to compare the two utterances. We used the same distance metric (#insertions + #deletions + #substitutions) used in word error rate. Unfortunately, distance measured in this way did not correlate at all with WER. This result makes sense in retrospect, since we would get a distance of 1 if the first utterance were “Seattle” and the second utterance were “Seattle, Washington”, “SeaTac” or silence. To better characterize types of user rephrasing, we would need hand labels, such as the “Add, Add/Omit, Omit, Paraphrase, Repeat” labels used by Swerts et al. [4].

### 4.2. Impact of System Situation on User Behavior

We have found two situations that seem important to avoid: users repeating themselves and user frustration. The first tends to make recognition worse. The second usually hampers recognition, but it is also worth avoiding for its own sake; user satisfaction is as much a goal as lower word error rate. We now look at how spiral depth and reprompt manner affect these two dimensions.

Spiral depth does not lead to a strict increase or decrease in user repeats compared to user rephrasings. As can be seen in Fig. 4, the fraction of user repeats seems to go up and down every two steps. Only the difference between depths 1 and 2 is significant, but the zigzag pattern agrees with Oviatt’s finding [3] that users frequently try something twice and then change their strategy. As is also shown in Fig. 4, people are more likely to be frustrated as they get deeper into a spiral error (not surprisingly). Here, single step distinctions are not significant, but the larger steps (1 vs. 4, 3 vs. 6) are significantly different.

The reprompt manner seems to affect frustration, as is shown in Figure 5. Apologizing is associated with a significantly lower rate of user frustration compared to not apologizing. The Not Understood category has a frustration rate similar

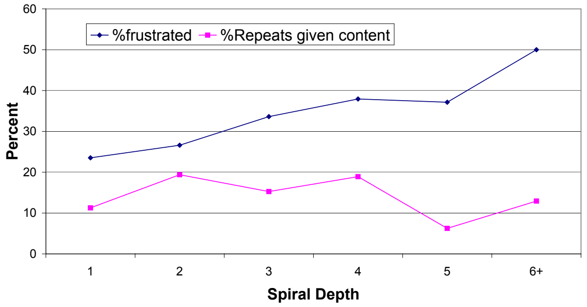

Figure 4: Rate of frustration utterances and user repeats at different spiral depths.

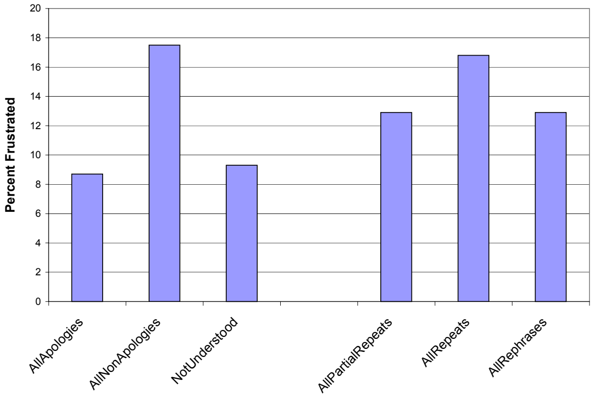

Figure 5: Percent frustrated after different repromptings.

to the apologies, which may in part be due to the fact that many of these prompts also included apologies. Since several systems use these types of prompts, reflecting a broad range of recognition WERs, the lower frustration rates cannot be explained by the fact that they were used in inherently "better" systems. Rephrasing and partial repeats are both associated with lower frustration rates than exact repeats. The patterns of apologies being useful is consistent across repeats and rephrasing, but not partial repeats.

The reprompt manner also had an effect on the user response. Although users are more likely to rephrase than to repeat their utterances after any prompt in an error subdialog, the frequency of rephrasing increases from 69% after an exact system repeat to 79% after a rephrased prompt and to 84% after a "not understood" prompt.

### 4.3. Impact of System Situation on WER

So far, we have demonstrated that frustration and the user's response manner affect WER, and that spiral depth and reprompt manner affect frustration and the user's response type. Thus, spiral depth and reprompt manner should affect WER. In this section, we demonstrate the extent of these effects.

As can be seen in Fig. 6, going deep into a spiral error definitely hampers recognizer performance. The differences between consecutive spiral depths are not generally significant. However, the increasing WER trend is definitely present at the lowest depths: the difference between spiral depths 1 and 3 is significant. Depths 6 and up are significantly worse than any specific lower depth.

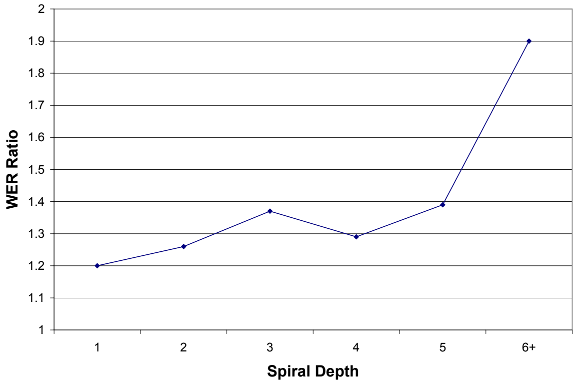

Figure 6: WER ratio of utterances at different spiral depths

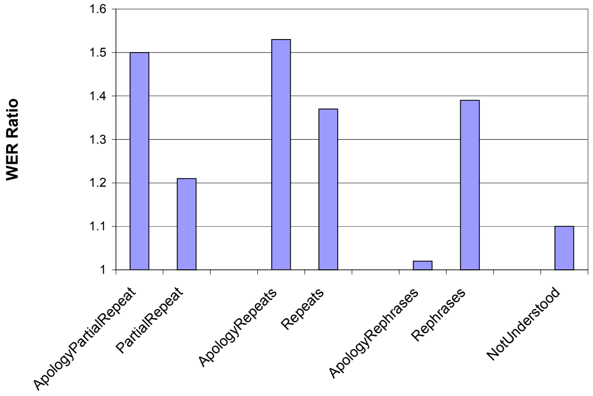

Figure 7: Reprompt Manner effects on WER Ratio

Many of the trends observed in the frustration effects of Response Manner correspond to similar patterns in word error differences, but not all, as shown in Fig. 7. The main conclusion – that the best system response strategy is to apologize and rephrase – also holds from the perspective of lowering the WER (or the WER ratio, since we are compiling statistics across systems). The WER ratio after Not Understood prompts is also lower. However, the WER ratio for all apologies is only a little lower than for non-apologies, and the WER ratio for apology repeats is actually higher than repeats without an apology. This difference relative to the frustration results cannot be explained by differences of spiral depth, so we cannot yet explain why the apologies did not have a positive effect on repetitions.

## 5. Discussion

Mixed-initiative dialog systems are error prone, and reprompts are extremely prevalent in current systems. Unfortunately, the deeper a spiral error goes, the more difficult it can be for the user to escape it, as their corrections and repetitions continue to be misrecognized. Other research has looked at detecting such error sub-dialogs and changing system initiative when errors occur. This paper looked at how the generator can be modified to improve error recovery and reduce user frustration. We report results of a corpus analysis aimed at identifying user behaviors that are correlated with higher vs. lower error rates, and at assessing whether system prompt wordings could influence these behaviors and hence impact error rates.

Utterances labeled as frustrated or annoyed were associated with significantly higher WER for all but 1 system, but only 1 system had significantly higher WER for utterances labeled as hyperarticulated (of 8 systems labeled, in both cases). The hyperarticulation result is not consistent with other reports, but it may reflect advances in ASR systems which now use much more training data (including hyperarticulated speech), and/or there could be differences in hyperarticulation annotation conventions (e.g. our annotation may have included a larger range of hyperarticulation, not just extreme cases). The fact that the results are different for hyperarticulation and frustration is plausible given that Ang et al. [17] found that the two characteristics are not highly correlated. In addition, user rephrases were found to have a significantly lower WER than exact repeats, so the natural tendency to rephrase is to be encouraged.

Not surprisingly, we found that increased spiral depth was associated with increased WER and increased percentage of utterances labeled as frustrated. We also found that, overall, responses with apologies were associated with a lower incidence of frustration. In that apologizing is a type of active support for emotion regulation, the results are consistent with conclusions in [19] that active support in response to frustration leads computer users to feel more positive about the system and able to continue the interaction longer. The “not understood” prompt – typically combined with an apology – also leads to lower frustration (though probably not if it is used repeatedly). Apologies led to lower WER when combined with rephrasing, but not for other reprompt types. Overall, we found that apologizing and rephrasing the system prompt rather than repeating it is associated with lower word error rates and lower frustration.

These results have implications for response generation modules and dialog control strategies which are relatively simple to implement. The findings generally match our intuitions: users are generally less frustrated if the error subdialogs are short and when the system apologizes. Perhaps more importantly, however, we find that the most common generation strategy of repeating a prompt verbatim is the worst tactic a system could take. Apologies combined with rephrasing are associated with lower WER in processing the user response, which can lead to faster error recovery. With access to the dialog manager’s internal state, reprompting (and hence need for an apology) is easily identified. The rephrasing generation strategy can be implemented in either stochastic or template generators, with the template generator simply requiring a list of possible rephrasings for each system prompt. Such changes should shorten human-computer dialogs and increase overall user satisfaction with the dialog system.

## Acknowledgments

This work was supported by DARPA grant no. N660019928924(02,04). The views contained in this document are those of the authors and should not be interpreted as reflecting the official policies of the funding agency.

## 6. References

[1] J. Shin, S. Narayanan, L. Gerber, A. Kazemzadeh, and D. Byrd, “Analysis of user behavior under error conditions in spoken dialogs,” in *Proc. ICSLP*, vol. 3, 2002, pp. 2069–2072.

[2] S. L. Oviatt and R. VanGent, “Error resolution during multimodal human-computer interaction,” in *Proc. ICSLP*, vol. 1, 1996, pp. 204–207.

[3] S. L. Oviatt, J. Bernard, and G. Levow, “Linguistic adaptation during error resolution with spoken and multimodal systems,” *Language and Speech*, vol. 41, pp. 419–442, 1998.

[4] M. Swerts, D. Litman, and J. Hirschberg, “Corrections in spoken dialogue systems,” in *Proc. ICSLP*, vol. I, 2000, pp. 254–257.

[5] M. Walker *et al.*, “DARPA Communicator dialog travel planning systems: the June 2000 data collection,” in *Proc. Eurospeech*, vol. 2, 2001, pp. 1371–1374.

[6] M. Eskenazi *et al.*, “Data collection and processing in the Carnegie Mellon Communicator,” in *Proc. Eurospeech*, 1999, pp. 2695–2698.

[7] S. L. Oviatt, P. R. Cohen, and M. Wang, “Toward interface design for human language technology: Modality and structure as determinants of linguistic complexity,” *Speech Communication*, vol. 15, pp. 283–300, 1994.

[8] G. Levow, “Characterizing and recognizing spoken corrections in human-computer dialogue,” in *Proc. ACL*, 1998.

[9] J. Hirasawa, N. Miyazaki, M. Nakano, and K. Aikawa, “New feature parameters for detecting misunderstandings in spoken dialogue systems,” in *Proc. ICSLP*, vol. II, 2000, pp. 154–157.

[10] J. Hirschberg, D. Litman, and M. Swerts, “Generalizing prosodic prediction of speech recognition errors,” in *Proc. ICSLP*, vol. II, 2000, pp. 615–618.

[11] K. Kirchhoff, “A comparison of classification techniques for the automatic detection of error corrections in human-computer dialogues,” in *Proc. NAACL Workshop on Adaptation in Dialogue Systems*, Pittsburgh, PA, 2001.

[12] J. Hirschberg, D. Litman, and M. Swerts, “Identifying user corrections automatically in spoken dialogue systems,” in *Proc. NAACL*, Pittsburg, PA, 2001.

[13] D. Litman, J. Hirschberg, and M. Swerts, “Predicting user reactions to system errors,” in *Proc. ACL*, 2001.

[14] M. A. Walker, J. Fromer, G. D. Fabbrizio, C. Mestel, and D. Hindle, “What can I say? Evaluating a spoken language interface to email,” in *Proc. CHI*, 1998, pp. 582–589.

[15] S. Narayanan *et al.*, “Effects of dialog initiative and multimodal presentation strategies on large directory information access,” in *Proc. ICSLP*, vol. II, 2000, pp. 636–639.

[16] S. Oviatt, G. Levow, E. Moreton, and M. MacEachern, “Modeling global and focal hyperarticulation during human-computer error resolution,” *JASA*, vol. 104, no. 5, pp. 3080–3098, 1998.

[17] J. Ang, R. Dhillon, A. Krupski, E. Shriberg, and A. Stolcke, “Prosody-based automatic detection of annoyance and frustration in human-computer dialog,” in *Proc. ICSLP*, 2002, pp. 2037–2040.

[18] H. Soltau and A. Waibel, “On the influence of hyperarticulated speech on recognition performance,” in *Proc. ICSLP*, vol. 2, 1998, pp. 229–332.

[19] J. Klein, Y. Moon, and R. Picard, “This computer responds to user frustration,” *Interacting with Computers*, vol. 14(2), pp. 119–140, 2002.