Predicting and Adapting to Poor Speech Recognition in a Spoken Dialogue System

Diane J. Litman
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932
diane@research.att.com

Shimei Pan
Computer Science Department
Columbia University
New York, NY 10027
pan@cs.columbia.edu

Abstract

Spoken dialogue system performance can vary widely for different users, as well for the same user during different dialogues. This paper presents the design and evaluation of an adaptive version of TOOT, a spoken dialogue system for retrieving online train schedules. Adaptive TOOT predicts whether a user is having speech recognition problems as a particular dialogue progresses, and automatically adapts its dialogue strategies based on its predictions. An empirical evaluation of the system demonstrates the utility of the approach.

Introduction

Most spoken dialogue systems do not try to improve performance by dynamically adapting the system's dialogue behaviors to an individual user during the course of a particular dialogue. But the performance of a spoken dialogue system can vary significantly for different users and even for the same user across dialogues. This paper presents the design and experimental evaluation of a spoken dialogue system that predicts and responds to problematic dialogue situations by automatically adapting its dialogue behaviors.

Figures 1 and 2 show two example dialogues with an adaptive version of TOOT, a spoken dialogue system for retrieving train schedules from the web.[1] In Figure 1, TOOT successfully acquires all of the information that it needs from the user within just one interaction. In Figure 2, however, the *same* system takes much longer to obtain the *same* information from a *different* user, due to repeated automatic speech recognition (ASR) errors (U1, U4, U5, U7).[2] Thus, TOOT's initial configuration of dialogue strategies (*user initiative* and *no confirmation*, as will be detailed below) seems well matched to the user in Figure 1 but poorly matched to the user in Figure 2. In fact, the second dialogue only starts to make real progress after TOOT (twice) adapts its behavior to use more conservative dialogue strategies. After the first adaptation, TOOT starts to preface its utterances with "I heard you say" (S5, S8), in order to *implicitly confirm* the most recent ASR hypothesis. When this adaptation still doesn't seem to help the user, the *system* takes the *initiative* by asking the user a series of directed questions (e.g., S9, S11, S13, S15), and asking the user to *explicitly confirm* each ASR hypothesis (e.g., S10, S12, S14, S16). After this second adaptation, the ASR misrecognitions are finally eliminated and the dialogue is successfully completed. Examples such as these suggest that it will be difficult to define a single "best" initial dialogue configuration for all users in all situations, and that a system should also be able to dynamically adapt its initial dialogue strategies.

S1:	Hi, this is AT&T Amtrak Schedule System. This is TOOT. How may I help you?
U1:	I want a train from Baltimore to New York on Sunday at 8:30 in the evening.
S2:	Do you want me to find the trains from Baltimore to New York on Sunday around 8:30 in the evening now?
U2:	Yes.
S3:	I am going to get the train schedules for you. This may take some time if the internet traffic is heavy. Please wait patiently. [pause] There are 2 trains available approximately at 8:30 in the evening. The first train leaves at 8:38 pm on Sunday, and it takes 2 hours 26 minutes. The second train leaves at 9:08 pm from Sunday to Thursday, and it takes 15 hours 7 minutes. Do you want to continue and find a new set of trains?
U3:	No.
S4:	Thank you for using AT&T Amtrak train schedule system. See you next time.

Figure 1: A dialogue where the system does not adapt.

In previous research, we demonstrated that the use of an *adaptable* spoken dialogue system can indeed increase system performance (Litman & Pan 1999). We developed an adaptable version of TOOT, where *users* were given instructions on how to use voice commands to change TOOT's dialogue strategies at any points in a dialogue. Results from an empirical evaluation showed that an adaptable version of TOOT outperformed a non-adaptable version. Unfortunately, a user-controlled adaptation system is not ideal for many applications, as it requires an initial (although mini-

[1]These examples are taken from the experimental corpus described below.

[2]The last column shows ASR confidence, as explained below. Note that the ASR interpretation ("ASR Output") is only shown for misrecognitions.

S1:	Hi, this is AT&T Amtrak Schedule System. This is TOOT. How may I help you?	
U1:	I need to find a train going to New York City. (*ASR Output: DC I don't care on Thursday.*)	-5.293949
S2:	Which city do you want to leave from?	
U2:	Baltimore.	-1.731395
S3:	Do you want me to find the trains from Baltimore to DC on Thursday at anytime now?	
U3:	No.	-2.095545
S4:	How may I help you?	
U4:	I need to leave on Sunday. (*ASR Output: DC between 1 night and 8.*)	-4.916628
*****	**TOOT automatically adapts its behavior to use a more conservative dialogue strategy.** *****	
S5:	I heard you say go to DC between 1 night and 8.	
	Do you want me to find the trains from Baltimore to DC on Thursday between 1 night and 8 now?	
U5:	No. (*ASR Output: Help.)*	-2.801172
S6:	You can say yes or no.	
U6:	No.	-2.509176
S7:	How may I help you?	
U7:	I need to go to New York City from Baltimore on Sunday at 8:30 pm.	
	(*ASR Output: To Chicago Tuesday 8:45 am.*)	-5.948381
S8:	I heard you say go to Chicago on Tuesday around 8:45 am.	
	Do you want me to find the trains from Baltimore to Chicago on Tuesday around 8:45 am now?	
U8:	No.	-3.355405
*****	**TOOT adapts its behavior again, to use an even more conservative dialogue strategy.** *****	
S9:	Which city do you want to go to?	
U9:	New York City.	-1.675092
S10:	Do you want to go to New York City?	
U10:	Yes.	-0.772970
S11:	Which city do you want to leave from?	
U11:	Baltimore.	-1.441275
S12:	Do you want to leave from Baltimore?	
U12:	Yes.	-0.609985
S13:	On which day of the week do you want to leave?	
U13:	Sunday.	-0.548220
S14:	Do you want to leave on Sunday?	
U14:	Yes.	-1.782579
S15:	What time do you want to leave?	
U15:	8:30 pm.	-1.727584
S16:	Do you want to leave around 8:30 pm?	
U16:	Yes.	-0.944121
S17:	Do you want me to find the trains from Baltimore to New York City on Sunday around 8:30 pm now?	
U17:	Yes.	-0.582485
S18:	I am going to get the train schedules for you . . .	

Figure 2: A dialogue where the same system infers that things are going poorly, and adapts its behavior twice.

mal) training session for users.

Thus, a natural extension of this approach is to design an *adaptive* spoken dialogue system, where the *system* rather than the user automatically controls the adaptation process. Recently, several machine learning approaches have been developed for detecting problematic dialogue situations that might warrant global dialogue strategy changes (Litman, Walker, & Kearns 1999; Walker *et al*. 2000). To date, however, none of these detection algorithms have actually been used to trigger an automatic adaptation process.

In this paper we show how to combine the above lines of research, by building an adaptive system based on the detection of a problematic dialogue situation, and empirically demonstrating its utility.[3] We first designed adaptive TOOT, which automatically infers and reacts to poor ASR performance in an online manner. We then conducted an experiment comparing the performance of adaptive TOOT to a comparable non-adaptive version. Our results showed that by adapting the dialogue strategies of TOOT in response to inferences regarding repeated ASR misrecognitions, we significantly improved the task success rate.

An Adaptive Spoken Dialogue System

We have developed both adaptive and non-adaptive versions of TOOT, a voice-enabled dialogue system for accessing train schedules from the web via a telephone conversation. TOOT is implemented using a spoken dialogue system platform that combines automatic speech recognition (ASR), text-to-speech synthesis (TTS), a phone interface, and modules for specifying a dialogue manager and application functions (Kamm *et al*. 1997). ASR in our platform is speaker-independent, grammar-based and supports barge-in (which allows users to interrupt the system). The dialogue manager uses a finite state machine to control the interaction, based

[3]See (Chu-Carroll & Nickerson 2000) for an evaluation of a spoken dialogue system that automatically adapts initiative based on the current *utterance* and dialogue history.

on the current system state and ASR results.

This section details our methodology for designing an adaptation component for use within the dialogue manager of the adaptive version of TOOT. First, we define the types of dialogue strategy choices that are allowed in TOOT. Second, we illustrate how we instantiate (Litman, Walker, & Kearns 1999) in order to learn a problematic dialogue classifier tailored for TOOT. Third, we describe the adaptation algorithm that we have developed which uses this classifier to predict and react to repeated ASR misrecognitions. Finally, we illustrate how this algorithm generates the dialogue behavior shown in Figure 2.

Dialogue Strategies for Initiative and Confirmation

We allow TOOT to use one of three possible initiative dialogue strategies ("system", "mixed" or "user") and one of three confirmation strategies ("explicit", "implicit", or "no"), at any point in a dialogue. The initiative strategy specifies who has control of the dialogue, while the confirmation strategy specifies how and whether the system lets the user know what it just understood.[4]

Consider the use of *user initiative with no confirmation*, the initial dialogue configuration used in Figures 1 and 2. This approach is the most natural approach in human-human conversation, and is feasible for human-machine conversations when the user knows what can be said at any points of a dialogue, and the system has good recognition performance for the user. By allowing users to specify any number of attributes in a single utterance and by not informing users of every potential misrecognition, this approach can lead to very short and effective dialogues, as in Figure 1.

In contrast, consider the use of *system initiative with explicit confirmation*, our most conservative parameterization of dialogue strategies. Although this configuration is cumbersome and typically increases total dialogue length (Walker, Fromer, & Narayanan 1998; Danieli & Gerbino 1995), it is sometimes effective as in the third portion of Figure 2. Giving the system the initiative about what to ask for next helps to reduce ASR misrecognitions (Walker, Fromer, & Narayanan 1998), by helping to keep the user's utterances within the system's vocabulary and grammar. The use of explicit confirmation also helps increase the user's task success (Danieli & Gerbino 1995), by making it easy for users to correct misrecognitions when they do occur.

A middle setting of dialogue strategies is illustrated in the second portion of Figure 2, where TOOT uses *mixed initiative with implicit confirmation*. In contrast to no confirmation, implicit confirmation makes the user aware of ASR errors; in contrast to explicit confirmation, it is more difficult for users to correct ASR errors after an implicit confirmation (Krahmer *et al.* 1999). In mixed but not system initiative mode, the system can ask both specific questions and open-ended questions (e.g., "How may I help you?"). However, in user but not in mixed initiative mode, the system will let the user ignore the specific questions (e.g., after the prompt "On which day of the week do you want to leave?", the user can say "I want a train at 8:00.")

In the non-adaptive version of TOOT, the initiative and confirmation strategies are specified once at the beginning of a dialogue, and cannot be changed until the next dialogue. To allow TOOT to dynamically change its strategies *within* a dialogue, we have augmented the non-adaptive version with a new adaptation component. Whenever the adaptation component predicts that repeated ASR problems have occurred during the course of a dialogue, it changes to a more conservative setting of dialogue strategies.

Learning to Detect Problematic Dialogues

One major functionality of the new adaptation component is that it needs to predict problematic situations during a dialogue, in order to trigger the dialogue strategy adaptations. Previous evaluations of a variety of spoken dialogue systems have suggested that ASR accuracy is one of the most significant predictors of dialogue system performance, e.g., (Walker, Fromer, & Narayanan 1998; Litman & Pan 1999). In our work, we have thus chosen to use poor ASR performance as our adaptation criterion. Following (Litman, Walker, & Kearns 1999), we employ a machine learning approach to automatically derive rules for classifying a dialogue as problematic with respect to ASR.

Our corpus consists of 120 dialogues collected from previous experiments with TOOT (Litman & Pan 1999). The dialogues illustrate many different dialogue strategy configurations, and were collected in interactions with novice users (undergraduate and graduate students). We first classify each dialogue in our corpus as "good" or "bad" with respect to ASR performance, by thresholding on the percentage of user utterances that were previously labeled as semantic misrecognitions.[5] Following (Litman, Walker, & Kearns 1999), our threshold is set to 11%, yielding 45 good dialogues and 75 bad dialogues. For example, the dialogue in Figure 1 would have been classified as "good" because there were no misrecognitions, while the portion of the dialogue in Figure 2 would have been classified as "bad" because 24% (4 out of 17) of the user utterances were misrecognitions.

We also extract a set of prediction features that represent high-level properties of the dialogue history, and that are automatically computable from the system log files generated for each dialogue. Again following (Litman, Walker, & Kearns 1999), we computed a set of 23 features that characterized dialogues along five dimensions: acoustic confidence, dialogue efficiency (e.g. number of system turns), dialogue quality or naturalness (e.g., number of user requests for help), experimental parameters (e.g., initial dialogue strategy configuration), and lexical (e.g. lexical items

[4] All other dialogue strategies (e.g., the response strategy for presenting the results of the web query) are fixed in advance, to control the factors in the experimental evaluation described below.

[5] The labeling of misrecognitions was done prior to the current research, by listening to the recordings and comparing them to the logged ASR results. When ASR did not correctly capture the task-related information, it was labeled as a misrecognition (e.g., U1, U4, U5, and U7 in Figure 2). Since the labeling is semantically based, if U9 had been recognized as "New York" then it still would have been labeled as a correct recognition. Although done manually, the labeling is based on objective criteria.

in ASR output). However, since (as will be seen below) our best learned ruleset uses only a single acoustic feature, only that feature is detailed here.

As shown in the last column of Figure 2, one source of acoustic information directly available in the system log is a per-utterance log-likelihood score from ASR, representing its "confidence" in its interpretation of the user's utterance (Zeljkovic 1996). These confidence measures are typically used to decide whether the system believes it has correctly understood the user's utterance. In our implementation, when the confidence score falls below a predefined threshold for each dialogue state, TOOT generates a rejection utterance such as "Sorry, I can't understand you. Please repeat your answer." The feature *predictedMisrecs%* (predicted percentage of misrecognized utterances) was derived from these utterance confidence scores as follows. First, a threshold of -4 was used to predict whether each non-rejected utterance in the dialogue was a misrecognition; second, the percentage of user utterances in the dialogue that corresponded to these *predicted* misrecognitions was computed. (Recall that our dialogue classifications of "good" and "bad" were determined by thresholding on the percentage of *actual* misrecognitions.) Thus for the excerpt in Figure 2, utterances U1, U4, and U7 would (correctly) be predicted as misrecognitions, and *predictedMisrecs%* would thus be 18% (3 out of 17 utterances). Note that U5 is (incorrectly) predicted to be a correct recognition.

Finally, once each dialogue in our corpus is represented in terms of its features and class value, we employ the machine learning program RIPPER (Cohen 1996) to automatically learn a poor ASR classification model from the training data. The classification model can be used to predict the class of future examples from their features, and is expressed as an ordered set of if-then rules. The best learned dialogue classifier for our data says that if the predicted percentage of misrecognitions is $>$ than 3%, then predict that the dialogue is "bad"; otherwise, predict "good".[6] Based on the results of 10-fold cross validation, this rule successfully classifies almost 80% of the dialogues in our corpus. This performance is better than a majority-class baseline (classify all dialogues as "bad") of 62%. The next section describes how we use this classification model in our adaptation component.

Predicting and Reacting to ASR Problems Online

Intuitively, the automatic adaptation component regularly monitors the conversation with respect to the features in the learned ruleset, and adapts to a more conservative dialogue strategy whenever the rules predict that the dialogue is having repeated ASR problems. The top portion of Figure 3 provides a pseudo-code sketch of the general adaptation algorithm, while the lower portion shows how we instantiate the system-dependent components of the algorithm for our experiments. In particular, the values of `AdaptFreq`, `Ruleset`, and `CurStrat`, as well as the algorithm for `MakeConservative(CurStrat)`, are specified at system initialization and represent parameters that potentially can be tuned to improve the performance of the algorithm.

```
Main
  ...
  specify adaptation frequency "AdaptFreq";
  specify classification model "Ruleset";
  specify initial strategy "CurStrat";
  for each user utterance
   if ((turns since CurStrat assignment) ≥ AdaptFreq)
    CheckRuleset(Ruleset);
  ...

CheckRuleset(Ruleset)
  for each rule R in Ruleset
   if (CheckPre(R) == "TRUE")
    if (RightHandSide(R) == "bad")
     AdaptStrategy(CurStrat);
    return;

AdaptStrategy(CurStrat)
  CurStrat ← MakeConservative(CurStrat);
-----------------------------------------------
AdaptFreq ← 4;

Ruleset ←
{if predictedMisrecs% > 3% then "bad";
default "good";}

(Initial) CurStrat:
 CurInit ← UserInit; CurConf ← NoConf;

MakeConservative(CurStrat)
  if (CurInit == UserInit) CurInit ← MixedInit
   elseif (CurInit == MixedInit) CurInit ← SystemInit;
  if (CurConf == NoConf) CurConf ← ImpConf
   elseif (CurConf == ImpConf) CurConf ← ExpConf;
```

Figure 3: Adaptation algorithm.

The system first checks the classification model `Ruleset` after the number of user utterances specified by `AdaptFreq`. In our implementation, `Ruleset` corresponds to the learned classification model described above, and `AdaptFreq` is set to 4 because humans took approximately 4 utterances on average to initiate adaptation in (Litman & Pan 1999). Note that although our rules were learned by analyzing full dialogues, our adaptation algorithm starts applying the rules after only 4 utterances.[7]

Since in general there is more than one rule in a classification model, `CheckRuleset(Ruleset)` sequentially checks the precondition of each rule until it finds the first rule that is applicable. (Recall that rules in RIPPER are or-

[6] While in this experiment RIPPER learned only a single rule and used only a single feature, when the same data was combined with data from two other spoken dialogue systems (Litman, Walker, & Kearns 1999), RIPPER learned 5 rules and used 7 of the 23 features.

[7] Although we have not investigated the impact this change would have made to the classification accuracy results described above, using the first few utterances rather than the whole dialogue to predict problematic situations did not seriously degrade classification accuracy in the experiments of (Walker *et al.* 2000).

dered. Thus if multiple if-then rules are applicable, the first rule in the ordering determines the class; if no if-then rules are applicable, the default rule is used.) When the first applicable rule is found, if the rule also classifies the dialogue as "bad", dialogue strategy adaptation will be triggered before processing the next user utterance. Otherwise, no adaptation is performed.

More specifically, in order to test the precondition of a rule, `CheckPre(R)` parses the system log file in order to compute the value for each prediction feature presented in the classification rule. Note that each time the features are computed, the system uses only the portion of the log file since the last adaptation (i.e., from the beginning of the dialogue only if there have been no adaptations), because only this part of the dialogue reflects the appropriateness of the current dialogue strategy. If the precondition of the rule is true when it is instantiated with the computed values `CheckPre(R) == ''TRUE''` and the rule gets fired; then, if the fired rule classifies the current dialogue status as `''bad''`, `AdaptStrategy(CurStrat)` is activated to change the value of the current dialogue strategy (`CurStrat`) to a more conservative one. Once a rule has been fired and the dialogue classified (and the strategy possibly adapted, depending on the value of the right hand side of the rule), the system continues the monitoring process as the dialogue progresses.

In our specific instantiation of the algorithm, only one feature is employed in the classification model (`predictedMisrecs%`). First, the system parses the log file to extract the ASR confidence score for each user utterance since the last adaptation. Following the definition of `predictedMisrecs%`, the system tests whether each confidence score is less than -4.0, and if so, categorizes the corresponding user utterance as a predicted misrecognition. Then it computes, among all the user utterances considered, the percentage of user utterances just predicted to be misrecognitions. Once `predictedMisrecs%` is calculated, `CheckPre(R)` checks whether this value is greater than 3% (the precondition of the first rule in `Ruleset`). If so, since the portion of the dialogue since the last adaptation is classified as "bad" (`RightHandSide(R) == ''bad''`), `AdaptStrategy(CurStrat)` is called. Note that `AdaptStrategy(CurStrat)` is not called when the if-then rule is not applicable, since the next and last rule will classify the dialogue as "good" by default. `AdaptStrategy(CurStrat)` in turn calls the simple version of `MakeConservative(CurStrat)` shown in Figure 3, which changes user initiative to mixed initiative and mixed initiative to system initiative. Similarly, no confirmation is always changed to implicit confirmation and implicit confirmation to explicit confirmation. Note that when the current dialogue strategy is already the most conservative one (system initiative and explicit confirmation), no further changes are possible.

Example

We now detail how the dialogue strategy adaptations in Figure 2 are automatically generated using the adaptation algorithm in Figure 3. In our experiments, TOOT is always initialized with the dialogue strategy configuration *user initiative with no confirmation*, because these are the most "natural" initiative and confirmation strategies in human-human conversation, and this configuration was shown to benefit most from *user*-controlled adaptation (Litman & Pan 1999).

Because of the user initiative setting, TOOT begins the dialogue in Figure 2 with the open question "How may I help you?" The user's response U1 is then misrecognized by ASR. Because of the no confirmation setting, TOOT does not confirm its interpretation of U1 but instead asks the user for a new piece of information (S2). The user thus doesn't realize the misrecognition until S3, when TOOT asks the user if it should query the web database. (Since this query is an expensive operation, TOOT *always* tells the user the values that will be used for the query – independently of the confirmation strategy.) Since the user now realizes that there was an earlier misrecognition, the user tells TOOT not to query the web (U3). In turn, this causes TOOT to again try to get the information it needs from the user (S4). Since the adaptation frequency is initialized to 4 (`AdaptFreq` in Figure 3), TOOT does nothing with respect to adaptation from U1-U3.

After U4, however, for the first time TOOT checks whether the current dialogue history satisfies the precondition of the adaptation condition, namely the first rule in `Ruleset` in Figure 3. First, TOOT calculates the value of `predictedMisrecs%` for the dialogue segment U1-U4. Because the ASR confidence scores for U1 and U4 are less than the threshold of -4.0, `predictedMisrecs%` is 50%. As a result, the adaptation rule is fired, the dialogue is classified as "bad", and TOOT adapted to a more conservative configuration of dialogue strategies (*mixed initiative with implicit confirmation*, following `MakeConservative(CurStrat)` in Figure 3).

After the first adaptation, the dialogue still doesn't go very well, as TOOT misrecognizes U5 and U7. After U8 (4 turns since the last CurStrat assignment), TOOT checks the classification model for the second time, but only with respect to these last 4 turns. That is because U5-U8 is the only portion of the dialogue obtained using the current strategies. Since the ASR confidence score for U7 is less than -4.0, `predictedMisrecs%` for the new dialogue segment is 25%. This value triggers another adaptation, this time to the most conservative configuration in our implementation (*system initiative with explicit confirmation*).

After this second adaptation, TOOT next checks the adaptation condition after U12 (for the dialogue history U9-U12). This time `predictedMisrecs%` is 0, so the default rule is applicable and no adaptation is triggered. (Given our simple `MakeConservative` algorithm, even if a third adaptation had been triggered, there would have been no more conservative strategies to switch to.) Also, unlike after U4 and U8, the number of turns since the last adaptation does not return to 0. TOOT thus continues to check the adaptation condition with each subsequent utterance (e.g., after U13 the relevant dialogue history is U9-U13), since `predictedMisrecs%` is always 0. Thus, after the second adaptation, the dialogue finally proceeds smoothly and the user's task is successfully completed.

Experimental Evaluation

In order to empirically verify that our automatic adaptation algorithm can actually improve spoken dialogue system performance, we evaluated the adaptive and non-adaptive versions of TOOT discussed in the previous sections. 6 novice users carried out 4 tasks with the adaptive version of TOOT, while 6 different novice users carried out the same 4 tasks with the previous non-adaptive version of TOOT. Our experimental corpus thus consisted of 48 dialogues.

Subjects for both versions of TOOT were undergraduate and graduate students from different universities. Subjects used the web to read a set of experimental instructions, then called TOOT from a phone. The experimental instructions included a brief description of TOOT's functionality, hints for talking to TOOT, and links to 4 task scenarios. The following task scenario was used for the dialogues in Figures 1 and 2: "Try to find a train going *to* **New York City** *from* **Baltimore** on **Sunday** at **8:30 pm**. If you cannot find an exact match, find the one with the **closest** departure time. Please write down the **exact departure time** of the train you found as well as the **total travel time**."

We used the data that we experimentally obtained to compute a number of measures relevant for spoken dialogue evaluation. First, the dialogue manager's log was used to automatically calculate measures representing the efficiency of the dialogue (e.g., total number of **system turns**). As discussed above, we also used each log and the corresponding dialogue recording to hand-label ASR misrecognitions. This allowed us to compute total number of **misrecognized user turns** per dialogue, a measure of dialogue quality. In addition, we manually computed an objective measure representing whether users successfully achieved their task goal or not (**task success**). Task success is 1 if both the **exact departure time** and the **total travel time** (written down by the user at the conclusion of the experiment) are correct, 0.5 if only one value is correct, and 0 if neither is correct. Finally, after each dialogue, users filled out a survey following (Litman & Pan 1999), where 8 questions measured usability factors. For example, "Did you know what you could say at each point of the dialogue?" measured perceived **User Expertise**, and answers from 1 to 5 were possible. A comprehensive **User Satisfaction** measure (ranging from 8 to 40) was then computed by summing each question's score.

We use analysis of variance (ANOVA) to determine whether the adaptive version of TOOT yields significant improvements for any of the evaluation measures used in our experiment.[8] As shown in Table 1, the adaptive version of TOOT outperforms the non-adaptive version.[9] From the

Evaluation Measure	Adaptive (n=24)	Non-Adaptive (n=24)	P
Task Success	.65	.23	.01
User Expertise	4	3.2	.09
User Satisfaction	25.6	21.6	.20
# of Misrecognized Turns	3.9	6.0	.15
# of System Turns	13.7	17.4	.28

Table 1: Dialogue means for different versions of TOOT.

means in column 2 and 3, adaptive TOOT on average has a higher task success rate, higher user satisfaction (particularly due to higher levels of feelings of expertise), less misrecognized user turns and less overall system turns. The P-value in column 4 indicates that the improvement in task success rate is significant for the adaptive version of TOOT ($P < 0.01$). In particular, task completion increases from 23% in the non-adaptive version to 65% in the adaptive version. This verifies that adaptation can significantly improve TOOT's performance, in our case by helping users to better achieve their task goals. The improvement for user expertise also shows a trend towards statistical significance ($P < 0.09$). More data is needed to see whether we can obtain significance for the improvements using the other metrics.

It is also interesting to more informally examine how adaptation varies across both dialogues and users. For the 24 dialogues with the adaptive version of TOOT, TOOT didn't adapt at all in 5 dialogues, and adapted at least once in the remaining 19 dialogues. Furthermore, breaking down the .65 overall task success rate (Table 1) by these two conditions shows that the average task success rate was .60 when TOOT chose not to adapt, and .66 when TOOT decided to adapt. Thus, adaptive TOOT does indeed seem to keep the initial dialogue strategy configuration only when appropriate (in contrast to the non-adaptive version of TOOT, where the success rate for the initial configuration is .23), and adapts otherwise. The frequency of adaptation also differs across subjects: for 3 subjects, TOOT adapted in all 4 dialogues; for 1 subject, TOOT adapted in 3 out of 4 dialogues; for the remaining 2 subjects, TOOT adapted for only 2 dialogues. It is particularly interesting to compare the only two subjects who successfully completed all 4 tasks. For one of

[8] Our experimental design consisted of 2 factors: the within-in group factor *system adaptability* (with values adaptive or non-adaptive) and the between-groups factor *task* (with values one through four). We use a two-way analysis of variance (ANOVA) to compute whether any main (task-independent) effects of adaptability are statistically significant (probability p<.05) or show a trend (probability p<.1). The ANOVA also tests whether there are any main effects of task, or any interaction effects between adaptability and task, but there are no such significant effects in our data.

[9] Our previous results showed that the utility of user-controlled adaptation was greatest for *user initiative with no confirmation* TOOT (Litman & Pan 1999). Hence, for our first attempt at automatic adaptation, we focused on adapting only this initial dialogue strategy. However, unpublished results from our previous experiment also showed that in addition to adaptability effects, there were main effects for a factor not considered here: initial dialogue strategy (*user initiative with no confirmation* versus *system initiative with explicit confirmation*). Users had both higher task completion and user satisfaction rates for 1) the adaptable versions of TOOT (independently of strategy), and 2) the *system initiative with explicit confirmation* versions of TOOT (independently of adaptability). In fact, *non*-adaptable *system initiative with explicit confirmation* TOOT achieved the same task success and the same user satisfaction as *user*-adaptable *user initiative with no confirmation* TOOT (although we speculate that the system initiative version would have done less well with a more expert user population).

these subjects TOOT always adapted twice, while for the other the number of adaptations was either 1 or 0, decreasing as the user gained experience. Observations such as these strengthen our belief that a fixed dialogue strategy will not be ideal for different users, and that even for the same user, different dialogue strategies may be needed in different circumstances.

Summary and Future Work

We have designed and implemented a fully-automated adaptive version of TOOT, and have empirically verified improved levels of system performance compared to a non-adaptive version. Our system incrementally predicts whether a user is having ASR problems as a dialogue progresses, and adapts to a more conservative set of dialogue strategies whenever the predictions classify the dialogue as problematic. Our experimental evaluation demonstrates that the adaptive system outperforms a non-adaptive version of the same system for novice users, by significantly increasing the task success rate from 23% to 65%.

We view our current results as a baseline demonstrating the utility of our approach, and hope to increase system performance by tuning the current implementation. For example, `MakeConservative` generates only two adaptations (even though many other initiative and confirmation configurations are possible, e.g. *user initiative with implicit confirmation*), and TOOT also can never switch back to a less conservative strategy. These types of sophisticated adaptation behaviors are observed when humans control adaptation (Litman & Pan 1999). We also want to optimize `AdaptFreq` by examining how our classifier's accuracy depends on the number of utterances used for prediction (Walker *et al.* 2000), and to explore the impact of using a sliding window rather than all the utterances since the last adaptation to compute `predictedMisrecs%`. Finally, while our current focus is on predicting and adapting to problems at the (sub)dialogue-level, we would like to apply our approach at the utterance level (Levow 1998; Litman, Hirschberg, & Swerts 2000; van Zanten 1999; Smith 1998; Chu-Carroll 2000).

Acknowledgments

We would like to thank Owen Rambow for commenting on an earlier version of this paper, and Sandra Carberry and Janyce Wiebe for their help in recruiting subjects. We especially thank the students at Columbia University, New Mexico State University, and the University of Delaware who participated in our experiment.

References

Chu-Carroll, J., and Nickerson, J. S. 2000. Evaluating automatic dialogue strategy adaptation for a spoken dialogue system. In *Proc. 1st Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Chu-Carroll, J. 2000. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. In *Proc. Applied Natural Language Processing (ANLP)*.

Cohen, W. 1996. Learning trees and rules with set-valued features. In *Proc. 13th National Conference on Artificial Intelligence (AAAI)*.

Danieli, M., and Gerbino, E. 1995. Metrics for evaluating dialogue strategies in a spoken language system. In *Proc. AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*.

Kamm, C.; Narayanan, S.; Dutton, D.; and Ritenour, R. 1997. Evaluating spoken dialog systems for telecommunication services. In *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*.

Krahmer, E.; Swerts, M.; Theune, M.; and Weegels, M. 1999. Error spotting in human-machine interactions. In *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*.

Levow, G.-A. 1998. Characterizing and recognizing spoken corrections in human-computer dialogue. In *Proc. 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL)*.

Litman, D. J., and Pan, S. 1999. Empirically Evaluating an Adaptable Spoken Dialogue System. In *Proc. 7th International Conference on User Modeling (UM)*.

Litman, D.; Hirschberg, J.; and Swerts, M. 2000. Predicting automatic speech recognition performance using prosodic cues. In *Proc. 1st Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Litman, D. J.; Walker, M. A.; and Kearns, M. J. 1999. Automatic detection of poor speech recognition at the dialogue level. In *Proc. 37th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Smith, R. W. 1998. An evaluation of strategies for selectively verifying utterance meanings in spoken natural language dialog. *International Journal of Human-Computer Studies* 48:627–647.

van Zanten, G. V. 1999. User modelling in adaptive dialogue management. In *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*.

Walker, M.; Langkilde, I.; Wright, J.; Gorin, A.; and Litman, D. 2000. Learning to predict problematic situations in a spoken dialogue system: Experiments with how may i help you? In *Proc. 1st Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Walker, M.; Fromer, J.; and Narayanan, S. 1998. Learning optimal dialogue strategies: A case study of a spoken dialogue agent for email. In *Proc. 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL)*.

Zeljkovic, I. 1996. Decoding optimal state sequences with smooth state likelihoods. In *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.