Designing and Evaluating an Adaptive Spoken Dialogue System

Diane J. Litman

University of Pittsburgh, Computer Science Department and LRDC, Pittsburgh, PA 15260 USA, litman@cs.pitt.edu

Shimei Pan

IBM T.J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532 USA, shimei@us.ibm.com

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. Based on rules learned from a set of training dialogues, adaptive TOOT constructs a user model representing whether the user is having speech recognition problems as a particular dialogue progresses. Adaptive TOOT then automatically adapts its dialogue strategies based on this dynamically changing user model. An empirical evaluation of the system demonstrates the utility of the approach.

Key words: adaptive spoken dialogue systems, hypothesis testing for the effectiveness of adaptations, PARADISE for evaluating performance measures, user model acquisition via machine learning, speech recognition

1. 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 applies learned rules for modeling whether a user is having speech recognition problems, and adapts its dialogue behaviors accordingly.

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

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

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.

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.

In previous research, we demonstrated that the use of an *adaptable* spoken dialogue system can indeed increase system performance (Litman and 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.

[2] The last column shows ASR confidence, as explained below. Note that the ASR interpretation ("ASR") 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: 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: DC between 1 night and 8*)	-4.916628
S5:	(*TOOT adapts*) 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: 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: 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
S9:	(*TOOT adapts*) 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 ...	
U17:	Yes ...	-0.582485

Figure 2. A dialogue where the same system adapts twice, based on user modeling.

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 minimal) 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 with users that might warrant global dialogue strategy changes (Litman et al., 1999; Walker et al., 2000a). To date, however, none of these algorithms for constructing a model of whether a user's interaction is problematic 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 spoken dialogue system based on a learned model of whether a user's interaction with the system is problematic, and empirically evaluating its utility. We first present the design of adaptive TOOT, which uses empirically learned rules to automatically infer and react to a model of user ASR problems in an online manner. We then present the design of an experiment which compares the performance of adaptive TOOT to a comparable non-adaptive version. The results of our experimental evaluation show that by adapting the dialogue strategies of TOOT based on a user model regarding repeated ASR misrecognitions, we significantly improve the task success rate. We also quantify the relative importance of multiple evaluation measures to performance.

2. 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 on the current system state and ASR results. The TOOT dialogue manager consists of 168 states, each of which is associated with one of 12 different grammars; these grammars specify the ASR language model at that point in the dialogue.

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 et al., 1999) in order to learn a problematic dialogue classifier from previous dialogues with TOOT, thus providing our user model with a strong empirical basis. Third, we describe the adaptation algorithm that we have developed which uses the learned user modeling component to predict and react to repeated ASR misrecognitions. Finally, we illustrate how our adaptation algorithm generates the dialogue behavior shown in Figure 2.

2.1. 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.[3]

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 et al., 1998a; Danieli and 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 et al., 1998a), 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 and Gerbino, 1995), by making it easy for users to correct misrecognitions when they do occur.

[3] 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.

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 the user is having repeated ASR problems during the course of a dialogue, the system changes to a more conservative setting of dialogue strategies.

2.2. Modeling Problematic User Dialogues: A Machine Learning Approach

One major functionality of the new adaptation component is that it needs to model whether a user is having problems 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 (Walker et al., 1998a; Litman and Pan, 1999), and that it is possible to improve ASR accuracy by changing dialogue strategies (Walker et al., 1997a). In our work, we have thus chosen to use poor ASR performance as our adaptation criterion. Following (Litman et al., 1999), we employ a machine learning approach to automatically derive rules for classifying a dialogue as problematic with respect to ASR. Given a set of features that can be automatically monitored during the course of a dialogue, our approach allows us to learn from training data which subset of features to incorporate into the user model, and rules for predicting problematic ASR based on these features.

Our corpus consists of 120 dialogues collected from previous experiments with TOOT (Litman and Pan, 1999). The dialogues illustrate many different dialogue strategy configurations, and were collected in interactions with novice users (undergraduate and graduate students). Prior to the current research, each utterance in our corpus was labeled for *semantic accuracy*, by listening to the recordings and comparing them to the logged ASR results. For example, if the user said "I want to go to Baltimore on Saturday at ten o'clock" but the result of ASR was "Go to Boston on Saturday", the semantic accuracy score for this turn would be .33. Furthermore, when the semantic accuracy score was less than 1, i.e. when ASR did not correctly capture the task-related information, the utterance was also labeled as a semantic *misrecognition* (e.g., the example just given, and also U1, U4, U5, and U7 in Figure 2). Note that 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. Also note that although the labeling was done manually, it was based on objective criteria.[4]

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. Following (Litman et al., 1999), our threshold for percentage of semantic misrecognitions in a dialogue is set to 11%, yielding 45 good dialogues and 75 bad dialogues.[5] 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 et al., 1999), we computed a set of 23 features that characterized dialogues along five dimensions: acoustic confidence, dialogue efficiency (e.g. number of system and user 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 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-

[4] While word accuracy would have been another way of evaluating ASR success, semantic accuracy is a more appropriate measure for dialogue adaptation, because it does not penalize for word errors that are unimportant to overall utterance interpretation.

[5] A threshold of 11% was used in (Litman et al., 1999) because it roughly balanced the classes in their corpus, which consisted of data from TOOT and two other spoken dialogue systems. 11% was also consistent with a threshold inferred from human judgements (Litman, 1998). Note, however, that this same threshold yields a less balanced class distribution for our current corpus.

likelihood score from ASR, representing its "confidence" in its interpretation of the user's utterance (Zeljkovic, 1996). These acoustic 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."

Unfortunately, the use of these confidence scores is not trivial, so the rejection process often either rejects correctly recognized utterances, or does not reject misrecognitions. On the one hand, there is no simple one-to-one mapping between low confidence scores and incorrect recognitions, and the setting of a rejection threshold is thus a matter of trial and error (Bouwman et al., 1999). On the other hand, the presence of word errors should not necessarily lead to a rejection, since some incorrect word recognitions do not necessarily lead to misrecognition at the semantic level that we are concerned with. The TOOT thresholds were set such that TOOT tended to incorrectly recognize utterances rather than incorrectly reject them, hence the need to detect when the user is experiencing a high level of misrecognitions.

To predict this type of situation, four versions of a feature called *predictedMisrecs%* (predicted percentage of misrecognized utterances) were derived from the utterance confidence scores as follows. First, a threshold (independent of dialogue state) was used to predict whether each non-rejected utterance in the dialogue was a misrecognition; thresholds used for the four versions of the feature were $-2, -3, -4, -5$, and were chosen by hand from the entire dataset to be informative. The four thresholds represent different (coarse) approximations to the distribution of log-likelihood scores in the dialogue. Note that unlike the confidence score thresholds, the *predictedMisrecs%* thresholds are used to predict misrecognition rather than rejection, and are fixed across all dialogues rather than being dependent on system state. The version of the feature learned in the results described below uses a threshold of -4, and thus predicts that if a non-rejected utterance has a confidence score below -4 then it is a misrecognition. Second, the percentage of user utterances in the dialogue that corresponded to these *predicted* misrecognitions was computed. (Recall that our dialogue classifications were determined by thresholding on the percentage of manually labeled *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 uses only the single feature *predictedMisrecs%*:

if (predictedMisrecs% $>$ 3%) **then** "bad"
default is "good"

The first rule says that if the percentage of user utterances that are predicted to be misrecognitions (using a confidence score threshold of -4) is greater than 3%, then classify the dialogue as "bad". The second rule says otherwise, classify the dialogue as "good".[6]

To evaluate our user modeling component's accuracy, the error rate of the learned ruleset is estimated using the resampling method of *cross-validation* (Weiss and Kulikowski, 1991). In 10-fold cross-validation, the total set of examples is randomly divided into 10 disjoint test sets, and 10 runs of the learning program are performed. Thus, each run uses the examples not in the test set for training and the remaining examples for testing. An estimated error rate is obtained by averaging the error rate on the testing portion of the data from each of the 10 runs. Based on the results of 10-fold cross validation, our learned ruleset 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.

2.3. 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 user 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

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

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 and Pan, 1999). Given our learned classification model, this means that the system will adapt if it detects at least one misrecognition in this window of 4. 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 ordered. 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.

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%

[7] Although we have not investigated the impact this change would have made to the classification accuracy results described above, using only the first two utterances rather than the full dialogue to predict problematic dialogues in the experiments of (Walker et al., 2000a) only degraded classification accuracy from 87% to 80%. However, it should be noted that (Walker et al., 2000a) classified dialogues as problematic with respect to task success, while our definition of problematic was based on ASR misrecognition.

(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.

2.4. 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 and 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` 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 the predicted misrecognition percent 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.

3. Experimental Design

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. Our experiment was designed to test if *adaptive* TOOT performed better than *non-adaptive* TOOT, and whether any differences depended on the user's task. Our design thus consisted of two factors: *adaptability* and *task scenario*. In particular, six users carried out four tasks with adaptive TOOT, while six different users carried out the same four tasks with non-adaptive TOOT. The four task scenarios are shown in Figure 4, and were performed in sequence. The dialogues in Figures 1 and 2 were generated by users perform-

Task 1: Try to find a train going *to* **Chicago** *from* **Baltimore** on **Saturday** at **8 o'clock in the morning**. 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**.

Task 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**.

Task 3: Try to find a train going *to* **New York City** *from* **Philadelphia** on **Sunday** at **10:30 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**.

Task 4: Try to find a train going *to* **Washington D.C.** *from* **New York City** on **Monday** at **9:30 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**.

Figure 4. Task scenarios.

ing Task 2. Our experiment yielded a corpus of 48 dialogues (727 user turns).

Subjects were twelve undergraduate and graduate students from different universities. Subjects were not involved with the design or implementation of TOOT, and were novice users of spoken dialogue systems in general. Six of the subjects were randomly assigned to adaptive TOOT and six to non-adaptive TOOT.

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 four task pages. Each task page contained one of the task scenarios shown in Figure 4, the hints, instructions for calling TOOT, and a web survey designed to ascertain whether the user solved the task and to measure user perceptions of system usability. The experimental instructions and the task page for scenario 2 are shown in the Appendix, while the web survey is described below.

We used the data that we experimentally obtained to compute a number of measures relevant for spoken dialogue evaluation. Following PARADISE (Walker et al., 1997b; Walker et al., 1998b), we organize our evaluation measures along four performance dimensions, as shown in Figure 5. First, by logging the dialogue manager's behavior on entering and exiting each state in the finite state machine, we automatically calculated the total number of **Turns** per dialogue. The log was also used to calculate the percentage of user turns per dialogue that were **Timeouts** (when the user doesn't say anything within a

- *dialogue efficiency*: Turns
- *dialogue quality*: Timeouts, Rejections, Misrecognitions
- *task success*: Task Success
- *system usability*: User Satisfaction (based on TTS Performance, ASR Performance, Task Ease, Interaction Pace, User Expertise, System Response, Expected Behavior, Future Use)

Figure 5. Evaluation measures.

- Was TOOT easy to understand in this conversation? (**TTS Performance**)
- In this conversation, did TOOT understand what you said? (**ASR Performance**)
- In this conversation, was it easy to find the schedule you wanted? (**Task Ease**)
- Was the pace of interaction with TOOT appropriate in this conversation? (**Interaction Pace**)
- In this conversation, did you know what you could say at each point of the dialogue? (**User Expertise**)
- How often was TOOT sluggish and slow to reply to you in this conversation? (**System Response**)
- Did TOOT work the way you expected it to in this conversation? (**Expected Behavior**)
- From your current experience with using TOOT, do you think you'd use TOOT regularly to access train schedules when you are away from your computer? (**Future Use**)

Figure 6. Usability survey.

specified time frame, TOOT provides suggestions about what to say), and ASR **Rejections** (when the confidence level of ASR is too low, TOOT asks the user to repeat the utterance). In addition, by listening to recordings of the dialogues and comparing them to the logged ASR results, we manually labeled whether or not each user turn was an ASR (semantic) misrecognition. This allowed us to compute the percentage of **Misrecognitions** per dialogue

Users also filled out a web survey after each dialogue. Users first specified the departure and travel times that they obtained via the dialogue. Given that there was a single correct train to be retrieved for each task scenario, this allowed us to manually compute 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. In addition, users responded to the questionnaire shown in Figure 6 after each dialogue, which was used to assess users' subjective evaluation of TOOT's performance. Each question measured a particular usability factor, e.g., **TTS Performance**. Responses ranged over n pre-defined values (e.g., *almost never, rarely, sometimes, often, almost always*), and were mapped to an integer in $1 \dots 5$ (with 5 representing optimal performance). A comprehensive **User Satisfaction** measure was then computed by summing each question's score, and thus ranged in value from 8 to 40. Questionnaire-based user satisfaction ratings have been frequently used in the spoken dialogue literature as an external indicator of system usability (Polifroni et al., 1992; Shriberg et al., 1992).

4. Evaluation Results

In Section 4.1, we use analysis of variance (ANOVA) (Cohen, 1995) to determine whether the adaptive version of TOOT yields significant improvements for any of the evaluation measures used in our experiment. In Section 4.2, we use the PARADISE evaluation framework (Walker et al., 1997b; Walker et al., 1998b) to understand which of our evaluation measures best predicts overall performance in TOOT.

4.1. Adaptability Effects

Recall that our mixed experimental design consisted of 2 factors: the within-in group factor *adaptability* (with values adaptive, non-adaptive) and the between-groups factor *task scenario* (with values one through four). Each of our evaluation measures is analyzed using a two-way ANOVA for these factors. The ANOVA computes whether any main (task-independent) effects of adaptability are statistically significant (probability $p<.05$) or show a trend (probability $p<.1$). As we will see, our ANOVAs demonstrate a *main effect of adaptability* for the task success dimension of performance and show a trend for the system usability dimension. The ANOVA also tests whether there are any main effects of task, or any interaction effects between adaptability and task. In contrast to main effects, interaction effects of adaptability are not independent of the task scenario (i.e., the effects of adaptability and task scenario are not additive). In our data, there are no significant main effects of task and no significant interaction effects.

Table I summarizes the dialogue means for each of our evaluation measures, for both adaptive and non-adaptive TOOT, and also presents several measures for analyzing these results.[8] The raw data from the experiment is available in Appendix C. As discussed above, the probability value is used to determine whether the differences in the means are statistically significant. The effect size (here measured using omega-squared) gives the percentage of the total variability that is due to changing the adaptability of the system, while the power represents the percentage of experiments that when replicated with the same design, number of subjects, and effect size, would produce results with the same significance (Chin, 2001). In the social sciences, effect sizes of .01, .06, and .15 are typically considered to be small, medium and large, respectively; for power, .8 is the level that experimental psychologists are told to aim for, while .5 is the more typical level reported in publications.

As shown in the table, our data shows a significant main effect of adaptability for **Task Success**, and a trend for perceived **ASR Performance** and **User Expertise**. From the means in column 2 and 3, we can see that adaptive TOOT on average has a higher task success rate than non-adaptive TOOT. 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 p-value in column 4 indicates that the improvement in the task success rate for the adaptive version of TOOT is statistically significant ($p = 0.01$). **Task Success** also has a large effect size, as well as a high power rating. In addition, the improvements for two of the usability measures - **ASR Performance** and **User Expertise** - show a trend towards statistical significance ($p \leq 0.1$). Users perceive ASR performance to be more accurate for adaptive TOOT, and also have a better idea of what they should say throughout the dialogues with adaptive TOOT. **ASR Performance** and **User Expertise** have medium effect sizes, although the latter measure has high power.

The adaptive version of TOOT also outperforms the non-adaptive version for most of our other measures, although these results are currently not statistically significant. With respect to dialogue quality, the percentage of user turns where 1) the user times out and says noth-

[8] As noted in Section 2.4, for our first attempt at automatic adaptation, we focused on adapting only the user initiative with no confirmation initial dialogue strategy. This was because in our previous work, which also considered the experimental factor initial dialogue strategy, the utility of user-controlled adaptation was greatest for *user initiative with no confirmation* TOOT (Litman and Pan, 1999). That is, our previous results showed both main effects of adaptability, and interaction effects of adaptability and initial dialogue strategy. We have since extended the scope of this previous work, and have again found the same pattern of main and interaction effects of adaptability (Litman and Pan, 2001). We have also found a few minor effects of initial dialogue strategy.

Table I. Dialogue means for adaptive and non-adaptive versions of TOOT.

Evaluation Measure	Adaptive (n=24)	Non-Adaptive (n=24)	Probability	Effect Size	Power
Task Success	.65	.23	**.01	**.17	**.92
Timeouts	.00	.03	.11	*.10	*.69
Rejections	.09	.10	.63	.02	.07
Misrecognitions	.30	.38	.34	.01	.23
Turns	27.3	34.1	.32	.02	.27
User Satisfaction	25.6	21.6	.20	*.07	*.54
TTS Performance	1.5	2.4	.19	*-.11	*.73
ASR Performance	3.2	2.4	*.10	*.08	*.58
Task Ease	3.5	4.3	.20	*-.07	*.53
Interaction Pace	4.0	3.8	.64	.00	.15
User Expertise	4.0	3.2	*.09	*.13	**.82
System Response	3.6	3.3	.56	.00	.15
Expected Behavior	3.0	2.0	.20	.05	.41
Future Use	2.5	1.9	.25	.03	.33

**Significant at a 95% confidence level ($p \leq .05$); *trend at a 90% confidence level ($p \leq .1$)
**Large effect size of $\geq .15$; * medium effect size of $\geq .06$
**Ideal power of $\geq .8$; *typical power of $\geq .5$

ing, 2) ASR rejects the user's utterance, and 3) ASR misrecognizes the user's utterance[9], are lower for adaptive TOOT. With respect to efficiency, the dialogues with adaptive TOOT are shorter, taking an average of 27 turns as opposed to 34 turns with non-adaptive TOOT. With respect to system usability, users of adaptive TOOT have higher levels of overall user satisfaction, perceive the pace of the interaction to be better, feel that the system responds more quickly and in a more predictable manner, and would be more likely to use the system regularly. Note that for most of these measures the power is low, suggesting that there may not have been enough subjects to determine whether the improvements were significant.

There are only two (usability) measures for which non-adaptive TOOT performs better than adaptive TOOT (again, however, these results are currently not statistically significant). In particular, users of non-adaptive TOOT feel that it is easier to both understand TOOT's

[9] The adaptive version of TOOT also increases the mean semantic accuracy score per utterances (recall Section 2.2), from 64% to 74%.

Table II. Number of system adaptations per dialogue (Adaptive TOOT).

	Task 1	Task 2	Task 3	Task 4
Subject 1	1	2	2	1
Subject 2	1	1		
Subject 3		2		1
Subject 4	2	2	2	2
Subject 5	2	2	1	1
Subject 6	1		2	2

utterances and to find the train schedules, compared to users of adaptive TOOT.

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, Table II shows the number of times that the system adapted per dialogue.[10] The table shows that TOOT didn't adapt at all in 5 dialogues, and adapted at least once in the remaining 19 dialogues. Furthermore, if we break down the .65 overall task success rate (Table I) by these two conditions, we find 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, and adapts otherwise. This is in contrast to the non-adaptive version of TOOT, where the success rate for the initial configuration is only .23 (Table I) . Table II also shows that the frequency of adaptation 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 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.

[10] Recall that given the system-dependent settings shown in Figure 3, the number of possible adaptations per dialogue ranges from only 0 (blank in our table) to 2.

4.2. Contributors to Performance

To quantify the relative importance of our multiple evaluation measures to performance, we use the PARADISE evaluation framework to derive a performance function from our data (Walker et al., 1997b; Walker et al., 1998b). The PARADISE model posits that performance can be correlated with a meaningful external criterion of usability such as User Satisfaction. PARADISE then uses stepwise multiple linear regression to model User Satisfaction from measures representing the performance dimensions of task success, dialogue quality, and dialogue efficiency:

$$\text{User Satisfaction} = \sum_{i=1}^{n} w_i * \mathcal{N}(measure_i)$$

Linear regression produces coefficients (i.e., weights w_i) describing the relative contribution of predictor factors in accounting for the variance in a predicted factor. In PARADISE, the task success and dialogue cost measures are predictors, while User Satisfaction is predicted.[11] The normalization function $\mathcal{N}$ guarantees that the coefficients directly indicate the relative contributions.

The application of PARADISE to the TOOT data shows that the most significant contributors to **User Satisfaction** (USat) are **Misrecognitions** (Misrecs), **Turns**, **Rejections** (Rejs), and **Timeouts**, respectively. In addition, PARADISE shows that the following performance function provides the best fit to our data, accounting for 71% of the variance in User Satisfaction:

$$\text{USat} = -.67\mathcal{N}(\text{Misrecs}) - .39\mathcal{N}(\text{Turns}) - .33\mathcal{N}(\text{Rejs}) - .19\mathcal{N}(\text{Timeouts})$$

Misrecognitions and **Turns** are significant at $p = 0$, **Rejections** at $p < .0003$, and **Timeouts** at $p < .032$. Our performance function demonstrates that TOOT performance (estimated using subjective usability ratings) can be best predicted using a weighted combination of objective measures of dialogue quality and dialogue efficiency. In particular, fewer misrecognized user utterances, fewer total turns, fewer rejected user utterances, and fewer user timeouts all contribute to increasing perceived performance in TOOT.

Our performance equation helps explain the lack of a significant main effect of adaptability for User Satisfaction in Table I. Recall that only our ANOVA for Task Success showed a significant adaptability effect. Our PARADISE analysis, however, shows that Task Success

[11] Linear regression does not assume that predictors are independent, only that they are not highly correlated (e.g., because correlations above .70 can affect the coefficients, deletion of redundant predictors is advised (Monge and Cappella, 1980)).

is *not* a significant factor in explaining the variance in User Satisfaction. On the other hand, while the efficiency and quality measures did not show significant adaptability effects, they are significant factors in explaining User Satisfaction. Interestingly, the current PARADISE analysis yields a very different set of performance predictors compared to our previous evaluations of TOOT and other spoken dialogue systems, where measures of task success and dialogue quality were the most important predictors (Litman et al., 1998; Walker et al., 1998a; Kamm et al., 1998; Walker et al., 2000b). Understanding the reason for these differences is an area for future work.

5. Current Directions

We view our current results as a baseline demonstrating the utility of our approach, and hope to increase future system performance by tuning the current implementation. First, our automatic adaptation algorithm is currently extremely simple. For example, `MakeConservative` generates only two adaptations, while many other initiative and confirmation configurations are possible (e.g. *user initiative with implicit confirmation*, *user initiative with explicit confirmation*, *mixed initiative with no confirmation*, ...). In addition, adaptive TOOT only adapts in one direction, and never adapts back to less conservative dialogue strategies. Also, adaptive TOOT always waits for four utterances before adapting again, while a different and/or changing adaptation frequency might be more optimal. All of these more sophisticated adaptation behaviors are in fact observed when humans control adaptation (Litman and Pan, 1999).

One plan for enhancing our system is to look to our previous work on user-controlled adaptation for guidance. Table III compares our current results with the comparable subset of results from our previous experiment with *user-adaptable* TOOT (Litman and Pan, 1999), in which users could switch to a different initiative and/or confirmation strategy using voice commands, as shown in the excerpt below:

...

U: Change strategy.

S: You are using the user initiative. Which initiative do you want to change to? System, mixed, or user?

U: System initiative please.

S: You are using the no confirmation strategy. Which confirmation strategy do you want to change to? No confirmation, implicit confirmation, or explicit confirmation?

U: Implicit please.

S: I heard you say ...

Table III. Dialogue means for adaptive and user-adaptable versions of TOOT.

Evaluation Measure	Adaptive (n=24)	User-Adaptable (n=24)	P
Task Success	.65	.90	.30
*Timeouts	.00	.02	.00
Rejections	.09	.10	.81
Misrecognitions	.30	.19	.16
Turns	27.3	39.1	.32
+User Satisfaction	25.6	30.2	.07
TTS Performance	1.5	2.2	.17
ASR Performance	3.2	3.5	.25
Task Ease	3.5	2.9	.24
+Interaction Pace	4.0	3.4	.09
User Expertise	4.0	3.7	.50
*System Response	3.6	2.1	.02
Expected Behavior	3.0	3.8	.20
*Future Use	2.5	3.7	.02

*significant at a 95% confidence level ($p \leq .05$)
+trend at a 90% confidence level ($p \leq .1$)

Surprisingly, even with its current limitations, adaptive TOOT performs statistically comparably to user-adaptable TOOT for most performance measures. For the case of **Timeouts** and **System Response**, adaptive TOOT even outperforms the user-adaptable version (and a trend is shown for **Interaction Pace**). Nonetheless, users are significantly more likely to use user-adaptable TOOT in the future (**Future Use**), and show a trend for preferring user-adaptable TOOT with respect to overall system usability (**User Satisfaction**). Our comparison thus suggests that we need to focus on making adaptive TOOT more usable in the future.

Other plans for improving our adaptation algorithm include trying to optimize the adaptation frequency, by examining how the classifier's accuracy depends on the number of utterances used for prediction (Walker et al., 2000a), specific users, and users' increasing experience over time. We also plan to explore the impact of using a sliding window rather than all the utterances since the last adaptation to compute `predictedMisrecs%`.

6. Related Work

While previous work in the area of spoken dialogue suggested that a user's repeated problems with a system might warrant global dialogue strategy changes, our work is the first that both fully automates and empirically evaluates such an approach. While (Litman et al., 1999) and (Walker et al., 2000a) had already shown how to learn rules for predicting problematic dialogue situations (poor speech recognition performance, and transfers to a human operator, respectively), neither of the learned rulesets were ever incorporated back into a working system or evaluated with respect to adaptation. Conversely, while (Litman and Pan, 1999) empirically demonstrated that globally adapting dialogue strategies could improve system performance, the issues involved in modeling whether a user's dialogues were problematic and using the user model in an automated adaptation component were bypassed, due to the use of a human-controlled adaptation process.

While our work focuses on predicting and adapting to problems at the (sub)dialogue-level, other research has focused on modeling user problems at the utterance level, and investigating the types of dialogue strategy changes that might be warranted after a single problematic utterance. An adaptive approach to regulating initiative is described but not evaluated in (van Zanten, 1999), where the system uses a more conservative reprompt whenever a model of the user has been changed in response to a previous system prompt and an ASR error. A machine-learning approach for designing a spoken dialogue system that automatically adapts initiative based on participant roles, features of the current utterance and dialogue history is presented in (Chu-Carroll, 2000). An empirical evaluation shows that the adaptive system outperforms a non-adaptive version in terms of usability and efficiency (Chu-Carroll and Nickerson, 2000). Strategies for dynamically deciding whether to confirm each user utterance during a task-oriented dialogue are evaluated in (Smith, 1998). Simulation results suggest that adaptation strategies can improve performance, especially when the system has greater initiative. Although not yet used for adaptation, machine learning approaches for predicting single ASR misrecognitions have been developed based on the analysis of acoustic-prosodic features (Levow, 1998; Litman et al., 2000a). Also, descriptive analyses have shown differences in content and duration of user responses to correct and incorrect system confirmations (Hirasawa et al., 2000), and potential uses of (hand-labeled) positive and negative user feedback (Bell and Gustafson, 2000).

Finally, in contrast to the above spoken dialogue work, which focuses on adaptation at the (sub)dialogue and utterance levels, reinforcement learning has been used to adapt to more optimal initial dialogue strategies after training on multiple dialogues (Levin and Pieraccini, 1997; Litman et al., 2000b).

The empirical evaluation of an adaptive interface in a commercial software system (Strachan et al., 1997) is also similar to our work. Adaptive and non-adaptive versions of the same system were evaluated in experiments with human users. Analysis of variance demonstrated that an adaptive interface based on minimal user modeling improved subjective user satisfaction ratings.

7. Summary

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. Using a user modeling component learned from training data, 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 user model predicts that the user's dialogue is problematic. By using analysis of variance to examine how a set of evaluation measures differ as a function of adaptability, we elaborate the conditions under which adaptability leads to greater performance. Our main result is that adaptive TOOT outperforms non-adaptive TOOT for novice users, by significantly increasing the task success rate from 23% to 65%. By using PARADISE to derive a performance function from data, we also show that measures of dialogue quality and dialogue efficiency best predict a user's overall satisfaction with TOOT.

Acknowledgments

This work was performed at AT&T Labs - Research. We would like to thank David Chin for his computation of the effect size and power measures, 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 summer students at AT&T, and the students at Columbia University, New Mexico State University, and the University of Delaware who participated in our experiment.

Authors' Vitae

Dr. Diane Litman:

University of Pittsburgh, Computer Science Department and LRDC, Pittsburgh, PA 15260 USA

In 2001, Dr. Litman joined the University of Pittsburgh as an Associate Professor of Computer Science, and as a Research Scientist in the Learning Research and Development Center. Previously, she was a technical staff member at AT&T Labs – Research (formerly Bell Laboratories), and also an Assistant Professor of Computer Science at Columbia University. She received her Ph.D. in Computer Science from the University of Rochester in 1986. Dr. Litman's research interests include computational linguistics, knowledge representation, plan recognition, and spoken language processing. Dr. Litman is currently the Book Review Editor of *User Modeling and User-Adapted Interaction* and the Chair of the North American Chapter of the Association for Computational Linguistics.

Shimei Pan:

IBM T.J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532 USA

Appendix

A. Experimental Instructions

TOOT (THE AT&T AMTRAK TRAINS SCHEDULE SYSTEM): TASKS AND SURVEYS

General Description

TOOT, the AT&T Amtrak train schedule system, is an experimental spoken dialogue system that allows you to access train schedules from the web via a telephone conversation. You will be asked to call TOOT to do four different tasks. You should try to do each task as efficiently as you can and avoid listening to messages unnecessarily. Please make brief notes about the train departure time as well as the total travel time when you listen to the schedules.

Instructions for calling TOOT can be found at each task scenario. Please read through the instructions before calling. At the end of the task (after you hang up the phone), there are a few brief questions for you to answer. Even if TOOT aborted before you could complete the task, please finish this survey and continue to the next task. Thanks for participating!

Hints for Using TOOT

- If you don't know what to say or don't understand what TOOT is doing, say **Help** to hear a help message.
- If TOOT misunderstands you, say **Cancel** to try your utterance again.
- If you wait too long to tell TOOT what to do, TOOT will tell you what you can say.
- You can interrupt TOOT at any time. For example, if you've heard enough or if you know what you want to do, you don't have to wait for TOOT to finish talking.
- If you don't hear everything when TOOT presents the train schedules, say **Repeat** to hear the schedules again.
- If you want to abort your current attempt at the task before finishing, say **I'm done here** to start the dialogue again.
- When you are finished with a task, say **Goodbye** to end the dialogue.

Task Scenarios

You have four tasks to try in this experiment. You should do one task at a time. After you finish each task, **hang up** the phone and finish the survey.

- TASK 1: Click here to try task one.
- TASK 2: Click here to try task two.
- TASK 3: Click here to try task three.
- TASK 4: Click here to try task four.

Thank you for participating in this experiment!

B. Task Page for Scenario 2

TASK 2 SCENARIO AND SURVEY

Scenario Description

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**.

Hints for Using TOOT

- If you don't know what to say or don't understand what TOOT is doing, say **Help** to hear a help message.
- If TOOT misunderstands you, say **Cancel** to try your utterance again.
- If you wait too long to tell TOOT what to do, TOOT will tell you what you can say.
- You can interrupt TOOT at any time. For example, if you've heard enough or if you know what you want to do, you don't have to wait for TOOT to finish talking.
- If you don't hear everything when TOOT presents the train schedules, say **Repeat** to hear the schedules again.
- If you want to abort your current attempt at the task before finishing, say **I'm done here** to start the dialogue again.
- When you are finished with a task, say **Goodbye** to end the dialogue.

Calling TOOT

- Wait for Diane Litman to call you and connect you to TOOT. Then go ahead and do the task
- Hang up and complete the survey below.
- Even if TOOT "spaces out" on you part way through and you are unable to complete the scenario, please complete the survey.

Task 2 User Survey

Please make sure your answers to the survey reflect this conversation with TOOT. Please try to answer all the questions, and when finished, click the submit button at the bottom of the field.

Please enter the answers you found below.

- Exact Departure Time:
- Total Travel Time:

... THE 8 SURVEY QUESTIONS IN FIGURE 6 ...

C. Raw Data

```
Adaptability,TaskScenario,User,TaskSuccess,Timeouts,Rejections,Misrecognitions,Turns,
UserSatisfaction,TTSPerformance,ASRPerformance,TaskEase,InteractionPace,UserExpertise,
SystemResponse,ExpectedBehavior,FutureUse
"Adaptive","ONE","User01",0.5,0.04761905,0.1,0.3,41,27,2,3,4,4,5,5,5,3
"Adaptive","ONE","User02",1,0,0.05,0.35,40,27,1,3,3,4,5,3,1,3
"Adaptive","ONE","User03",0.5,0,0.1666667,0,12,39,1,5,1,4,5,1,5,5
"Adaptive","ONE","User04",0,0,0.4,0.4,10,18,2,1,5,3,4,5,1,1
"Adaptive","ONE","User05",1,0,0.05555556,0.3333333,36,21,1,3,4,4,3,4,1,1
"Adaptive","ONE","User06",0,0,0,0.5,12,22,1,2,2,5,3,4,1,3
"Adaptive","TWO","User01",1,0,0.1111111,0.3333333,36,24,2,3,4,5,4,5,5,3
"Adaptive","TWO","User02",1,0,0,0.3043478,46,26,4,3,4,4,5,4,5,3
"Adaptive","TWO","User03",1,0,0,0,6,37,1,5,1,4,5,3,5,5
"Adaptive","TWO","User04",1,0,0,0.2222222,36,26,2,4,4,2,3,4,5,3
"Adaptive","TWO","User05",1,0,0.1111111,0.1111111,18,25,1,3,2,4,4,3,1,1
"Adaptive","TWO","User06",0,0,0.2608696,0.3478261,46,16,1,1,5,5,3,4,1,1
"Adaptive","THREE","User01",0.5,0,0,0.4285714,14,27,1,4,3,5,4,5,5,3
"Adaptive","THREE","User02",1,0,0,0.3684211,38,26,4,3,4,4,5,4,5,3
"Adaptive","THREE","User03",1,0,0.1052632,0.2105263,38,23,1,2,4,4,5,3,1,1
"Adaptive","THREE","User04",0,0,0,0.5,8,25,1,4,5,3,4,2,1,1
"Adaptive","THREE","User05",1,0,0.1111111,0.1111111,18,32,1,4,2,4,4,3,5,3
"Adaptive","THREE","User06",1,0,0.2592592,0.1481482,54,18,1,4,5,5,3,5,1,1
"Adaptive","FOUR","User01",0,0,0.1,0.8,20,15,2,1,5,5,4,5,1,1
"Adaptive","FOUR","User02",1,0,0.05,0.25,40,28,2,3,3,4,4,4,5,3
"Adaptive","FOUR","User03",0,0,0,0.2222222,36,30,1,3,4,4,5,3,5,3
"Adaptive","FOUR","User04",0,0,0,0.6666667,12,21,1,2,5,3,1,1,1,1
"Adaptive","FOUR","User05",1,0,0,0,22,39,1,5,1,3,5,2,5,5
"Adaptive","FOUR","User06",1,0,0.25,0.25,16,23,1,5,3,5,3,5,1,3
"NonAdaptive","ONE","User07",0,0,0.1428571,0.5,28,23,3,3,5,4,3,3,5,1
"NonAdaptive","ONE","User08",0,0.07692308,0.08333334,0.25,25,20,3,3,4,4,2,4,1,3
"NonAdaptive","ONE","User09",1,0.1052632,0.0625,0.5625,35,13,4,2,4,5,1,4,1,1
"NonAdaptive","ONE","User10",0,0.05555556,0.05882353,0.5294118,35,18,5,2,5,4,4,4,1,3
"NonAdaptive","ONE","User11",0,0,0.2777778,0.4444444,37,21,1,2,5,4,3,2,1,1
"NonAdaptive","ONE","User12",1,0,0.1428571,0.1904762,42,31,1,3,4,3,4,2,5,3
"NonAdaptive","TWO","User07",0,0,0.04761905,0.2380952,42,19,4,3,4,4,3,3,1,1
"NonAdaptive","TWO","User08",0.5,0,0,0.2,20,28,2,3,3,4,3,3,5,3
"NonAdaptive","TWO","User09",0,0,0.2777778,0.3888889,36,14,3,1,5,5,4,5,1,1
"NonAdaptive","TWO","User10",0,0.03846154,0.08,0.4,51,18,5,2,5,4,4,4,1,3
"NonAdaptive","TWO","User11",0,0,0.2272727,0.3181818,45,16,1,1,5,4,2,5,1,1
"NonAdaptive","TWO","User12",0,0,0.04166667,0.1666667,49,24,1,3,5,3,4,2,1,1
"NonAdaptive","THREE","User07",1,0,0.1111111,0,18,35,2,4,2,3,4,2,5,5
"NonAdaptive","THREE","User08",0,0,0.25,0,8,34,2,4,1,3,4,2,5,3
"NonAdaptive","THREE","User09",0,0.1578947,0,0.6,34,15,2,1,5,5,4,5,1,1
"NonAdaptive","THREE","User10",0,0.09090909,0.2,0.6,21,20,4,3,5,4,4,4,1,3
"NonAdaptive","THREE","User11",1,0,0.1891892,0.2162162,75,17,1,1,5,4,2,4,1,1
"NonAdaptive","THREE","User12",0,0,0.02564103,0.3333333,79,22,1,3,5,3,3.6,3,1,1
"NonAdaptive","FOUR","User08",1,0,0,0.25,8,36,1,5,1,3,3,1,5,3
"NonAdaptive","FOUR","User09",0,0,0.05555556,0.3888889,36,13,3,1,4,5,2,5,1,1
"NonAdaptive","FOUR","User10",0,0.09090909,0.1,0.3,21,20,4,3,5,4,4,4,1,3
"NonAdaptive","FOUR","User11",0,0,0,1,6,19,1,2,5,4,2,3,1,1
"NonAdaptive","FOUR","User12",0,0,0.08333334,0.625,49,21,1,2,5,3,3,3,1,1
"NonAdaptive","FOUR","User07",0,0,0,0.5555556,18,21,2,1,5,3,4,2,1,1
```

References

Bell, L. and J. Gustafson: 2000, 'Positive and Negative User Feedback in a Spoken Dialogue Corpus'. In: *Proc. 6th International Conference of Spoken Language Processing (ICSLP)*. Beijing, China.

Bouwman, A. G., J. Sturm, and L. Boves: 1999, 'Incorporating confidence measures in the Dutch train timetable information system developed in the ARISE project'. In: *Proc. International Conference on Acoustics, Speech and Signal Processing*, Vol. 1. Phoenix, pp. 493–496.

Chin, D. N.: 2001, 'Empirical Evaluation of User Models and User-Adapted Systems'. *User Modeling and User-Adapted Interaction*.

Chu-Carroll, J.: 2000, 'MIMIC: An Adaptive Mixed Initiative Spoken Dialogue System for Information Queries'. In: *Proc. Applied Natural Language Processing (ANLP)*.

Chu-Carroll, J. and J. S. Nickerson: 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)*.

Cohen, P.: 1995, *Empirical Methods for Artificial Intelligence*. MIT Press, Boston.

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

Danieli, M. and E. Gerbino: 1995, 'Metrics for Evaluating Dialogue Strategies in a Spoken Language System'. In: *Proc. AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*.

Hirasawa, J., N. Miyazaki, M. Nakano, and K. Aikawa: 2000, 'New Feature Parameters For Detecting Misunderstandings in a Spoken Dialogue System'. In: *Proc. 6th International Conference of Spoken Language Processing (ICSLP)*, Vol. 2. Beijing, China, pp. 154–157.

Kamm, C., D. Litman, and M. Walker: 1998, 'From Novice to Expert: The Effect of Tutorials on User Expertise with Spoken Dialogue Systems'. In: *Proc. ICSLP*.

Kamm, C., S. Narayanan, D. Dutton, and R. Ritenour: 1997, 'Evaluating Spoken Dialog Systems for Telecommunication Services'. In: *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*.

Krahmer, E., M. Swerts, M. Theune, and M. Weegels: 1999, 'Error Spotting in Human-Machine Interactions'. In: *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*.

Levin, E. and R. Pieraccini: 1997, 'A Stochastic Model of Computer-Human Interaction for Learning Dialogue Strategies'. 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. Hirschberg, and M. Swerts: 2000a, '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., S. Pan, and M. Walker: 1998, 'Evaluating Response Strategies in a Web-Based Spoken Dialogue Agent'. In: *Proc. ACL/COLING*.

Litman, D. J.: 1998, 'Predicting Speech Recognition Performance from Dialogue Phenomena'. Presented at the American Association for Artificial Intelligence Spring Symposium Series on Applying Machine Learning to Discourse Processing.

Litman, D. J., M. S. Kearns, S. Singh, and M. A. Walker: 2000b, 'Automatic Optimization of Dialogue Management'. In: *Proc. of COLING 2000*.

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

Litman, D. J. and S. Pan: 2001, 'Evaluating User-Adaptability and Dialogue Strategy in a Spoken Dialogue System'. Manuscript in preparation.

Litman, D. J., M. A. Walker, and M. J. Kearns: 1999, 'Automatic Detection of Poor Speech Recognition at the Dialogue Level'. In: *Proc. 37th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Monge, P. and J. Cappella (eds.): 1980, *Multivariate Techniques in Human Communication Research*. Academic Press, New York.

Polifroni, J., L. Hirschman, S. Seneff, and V. Zue: 1992, 'Experiments in Evaluating Interactive Spoken Language Systems'. In: *Proc. DARPA Speech and NL Workshop*.

Shriberg, E., E. Wade, and P. Price: 1992, 'Human-Machine Problem Solving Using Spoken Language Systems (SLS): Factors Affecting Performance and User Satisfaction'. In: *Proc. DARPA Speech and NL Workshop*.

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.

Strachan, L., J. Anderson, M. Sneesby, and M. Evans: 1997, 'Pragmatic User Modelling in a Commercial Software System'. In: *Proc. UM97*.

van Zanten, G. V.: 1999, 'User Modelling in Adaptive Dialogue Management'. In: *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*.

Walker, M., J. Fromer, and S. Narayanan: 1998a, '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)*.

Walker, M., D. Hindle, J. Fromer, G. D. Fabbrizio, and C. Mestel: 1997a, 'Evaluating Competing Agent Strategies for a Voice Email Agent'. In: *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*.

Walker, M., I. Langkilde, J. Wright, A. Gorin, and D. Litman: 2000a, 'Learning to Predict Problematic Situations in a Spoken Dialogue System: Experiments with How May I Help You?'. In: *Proceedings of the North American Meeting of the Association for Computational Linguistics*.

Walker, M., D. Litman, C. Kamm, and A. Abella: 1997b, 'PARADISE: A General Framework for Evaluating Spoken Dialogue Agents'. In: *Proc. ACL/EACL*.

Walker, M., D. Litman, C. Kamm, and A. Abella: 1998b, 'Evaluating Spoken Dialogue Agents with PARADISE: Two Case Studies'. *Computer Speech and Language*.

Walker, M. A., C. A. Kamm, and D. J. Litman: 2000b, 'Towards Developing General Models of Usability with PARADISE'. *Natural Language Engineering: Special Issue on Best Practice in Spoken Dialogue Systems*.

Weiss, S. M. and C. Kulikowski: 1991, *Computer Systems That Learn: Classification and Prediction Methods from Stat istics, Neural Nets, Machine Learning, and Expert Systems*. San Mateo, CA: Morgan Kaufmann.

Zeljkovic, I.: 1996, 'Decoding Optimal State Sequences with Smooth State Likelihoods'. In: *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.