Automatic Evaluation: Using a DATE Dialogue Act Tagger for User Satisfaction and Task Completion Prediction

Helen Wright Hastie, Rashmi Prasad, Marilyn Walker

AT& T Labs - Research
180 Park Ave, Florham Park, N.J. 07932, U.S.A.
{hhastie, rjprasad, walker@research.att.com}

Abstract

The objective of the DARPA Communicator project is to support rapid, cost-effective development of multi-modal speech-enabled dialogue systems with advanced conversational capabilities. During the course of the Communicator program, we have been involved in developing methods for measuring progress towards the program goals and assessing advances in the component technologies required to achieve such goals. Our goal has been to develop a lightweight evaluation paradigm for heterogeneous systems. In this paper, we utilize the Communicator evaluation corpus from 2001 and build on previous work applying the PARADISE evaluation framework to establish a baseline for fully automatic system evaluation. We train a regression tree to predict User Satisfaction using a random 80% of the dialogues for training. The metrics (features) we use for prediction are a fully automatic Task Success Measure, Efficiency Measures, and System Dialogue Act Behaviors extracted from the dialogue logfiles using the DATE (Dialogue Act Tagging for Evaluation) tagging scheme. The learned tree with the DATE metrics has a correlation of 0.614 (R^2 of 0.376) with the actual user satisfaction values for the held out test set, while the learned tree without the DATE metrics has a correlation of 0.595 (R^2 of 0.35).

1. Introduction

The objective of the DARPA Communicator project is to support rapid, cost-effective development of multi-modal speech-enabled dialogue systems with advanced conversational capabilities. During the course of the Communicator program, we have been involved in developing methods for measuring progress towards the program goals and assessing advances in the component technologies required to achieve such goals. Our goal has been to develop an evaluation paradigm that supports continuous, lightweight, data collection and evaluation for heterogeneous systems. We have carried out two evaluation experiments within the Communicator program, one in June of 2000 resulting in 662 dialogues from 9 different Communicator travel planning systems, and a second evaluation carried out over six months in 2001, resulting in 1242 dialogues.

One problem with evaluation is that it is extremely costly. It often involves recruiting paid subjects to participate in dialogues with the system. In addition to carrying out some real or fixed tasks in dialogue with the system, subjects may be required to fill out a user profile and a user satisfaction survey, answer questions about task completion, or provide comments about the system's performance. The dialogues must be transcribed and some features of the interaction hand-labelled, such as aspects of the user's behavior, the task type, and task completion or reasons for no completion.

In this paper, we build on previous work applying the PARADISE evaluation framework to examine whether information useful for evaluation can be extracted from a corpus of dialogues using totally automatic means (Walker et al., 2000; Walker et al., 2002). It has been shown that dialogue acts can be useful for evaluation (Cattoni et al., 2001). Our work relies on an automatic dialogue act tagger DATE (Dialogue Act Tagging for Evaluation), that we developed for the Communicator domain that achieves 98.4% coverage and 96% accuracy on system utterances (Walker et al., 2001; Prasad and Walker, 2002). We experiment with using dialogue act labels in combination with other features as predictors of task completion (TaskCompletion) and user satisfaction (UserSatisfaction). We achieve 85% accuracy for predicting TaskCompletion; the UserSatisfaction predictor achieves a correlation of .61 with actual UserSatisfaction values (R^2 of 0.37) in a held out test set.

Section 2. briefly summarizes the PARADISE framework and describes our novel application of PARADISE in this work. Section 3. describes the experimental corpus. Section 4. presents the DATE dialogue act tagger which is used as the primary source of features for the automatic UserSatisfaction predictor. Section 5. describes how we extract a feature for automatically predicting TaskCompletion. Section 6. describes the experimental design for UserSatisfaction prediction and Section 7. presents the prediction results. We postpone the discussion of previous work until Section 8. for comparison purposes and present the conclusion and future developments in Section 9..

2. PARADISE Evaluation Framework

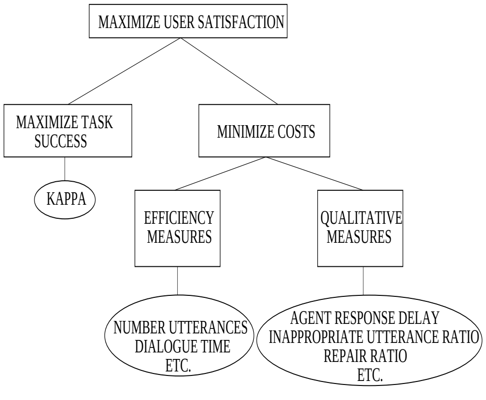


Figure 1: PARADISE's structure of objectives for spoken dialogue performance

The PARADISE evaluation framework uses methods from decision theory (Keeney and Raiffa, 1976; Doyle, 1992) to combine a disparate set of performance measures (i.e., user satisfaction, task success, and dialogue cost, all of which have been previously noted in the literature) into a single performance evaluation function (Walker et al., 2000). The use of decision theory requires a specification of both the objectives of the decision problem and a set of measures (known as attributes in decision theory) for operationalizing the objectives. The PARADISE model is based on the structure of objectives shown (in rectangles) in Figure 1; it posits that performance can be correlated with a meaningful external criterion such as usability, and thus that the overall goal of a spoken dialogue agent is to maximize an objective related to usability. UserSatisfaction ratings (Kamm, 1995; Shriberg et al., 1992; Polifroni et al., 1992) have been frequently used in the literature as an external indicator of the usability of a dialogue agent. The model further posits that two types of factors are potential relevant contributors to UserSatisfaction (namely Task Success and Dialogue Costs), and that two types of factors are potential relevant contributors to costs (namely Efficiency Measures and Dialogue Quality Measures).

PARADISE has been broadly applied in previous work (Walker et al., 2000; Lamel and Rosset, 2000; Bonneau-Maynard et al., 2000). Several uses have been made of the models derived by applying PARADISE. First, the overall performance metric can be used to automatically train the dialogue manager (Walker, 2000). Second, if metrics representing dialogue strategies are included in the Dialogue Quality Measures, then the significant predictors of UserSatisfaction can indicate which dialogue strategies are optimal. In our work, the purpose of the DATE tagging scheme is to extract such metrics (Walker et al., 2001; Prasad and Walker, 2002). Third, the models predict to what extent improvements in system components will increase UserSatisfaction. For example, if ASR (automatic speech recognition) performance has a +.25 correlation with UserSatisfaction in a standardized model, the prediction is that each unit change in ASR performance will result in a .25 unit increase in user satisfaction.

Our approach differs from previous work applying PARADISE in several respects. First, previous work has used both hand-labelled and automatically extracted metrics, but we look at utilizing only fully automatic metrics to explore the potential of fully automatic evaluation of dialogue systems. We believe there are a number of applications for a module that can predict UserSatisfaction automatically. For example, such a prediction can be used for deciding which dialogues in a large corpus are worth transcribing, or it could be factored into the dialogue manager and ASR modules to support online adaptation of the system. Second, rather than linear models, we apply Classification and Regression Trees (CART) to the prediction of UserSatisfaction (Brieman et al., 1984).

3. Experimental Corpus

The corpus used in these experiments is a corpus of 1242 dialogues collected for a Communicator evaluation experiment during six months of 2001. Three types of tasks are represented in the corpus:

- 350 Complex Trips (multiple legs and car, hotel arrangements)
- 694 Real Trips, of user's choice
- 198 Round Trips

Eight different systems participated in the evaluation. All sites implemented a logfile standard supporting a standard set of dialogue metrics such as number of system and user turns, dialogue duration, and time spent in each system module. The sites also provided both ASR and hand-labelled transcriptions for each user utterance.

On completion of each dialogue, the user was asked to fill-out a survey indicating the user's satisfaction (UserSatisfaction) with the system and perception of TaskCompletion. UserSatisfaction is calculated by summing the degree of the user's agreement on a five point Likert scale to five statements about the systems performance: (1) In this conversation, it was easy to get the information that I wanted (TaskEase); (2) I found the system easy to understand in this conversation (TTSPerf); (3) In this conversation, I knew what I could say or do at each point of the dialogue (UsrExpertise); (4) The system worked the way I expected it to in this conversation (ExpectedBehavior); (5) Based on my experience in this conversation using this system to get travel information, I would like to use this system regularly (FutureUse). Figure 2 gives the distribution of UserSatisfaction for the three types of trips showing that the complex tasks resulted in lower UserSatisfaction.

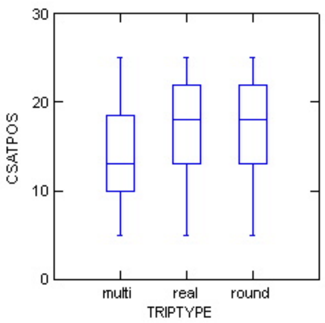


Figure 2: UserSatisfaction by TripType

The user's perception of TaskCompletion and the task requirements are used to define a ternary TaskCompletion metric: 0 indicates task failure; 1 indicates completion of an airline itinerary; 2 indicates completion of both airline and car/hotel arrangements. We also defined a Binary TaskCompletion metric where task failure is 0 and any level of TaskCompletion (airline and optionally car/hotel) is 1.

The goal of our experiments is to train and test a fully automatic predictor of UserSatisfaction using this evaluation corpus. The training set consists of a random 80% of the dialogues (994 dialogues) and the test set the remaining 20% (248 dialogues). We apply CART to this problem using features extracted from the logfiles as predictive features and the UserSatisfaction metric described above as the response variable. There are two types of input features whose extraction from the original logfiles required substantive work. First, as described above, we label all the system utterances in the logfiles with the DATE dialogue act tagging scheme. Secondly, we train a fully automatic TaskCompletion predictor whose output can be used as a fully automatic input feature for the prediction of UserSatisfaction. Section 4. first describes how we label the dialogues with the DATE tags and Section 5. describes how we use the dialogue act tags to predict TaskCompletion, before describing the training of the UserSatisfaction predictor in more detail in Section 6..

4. Dialogue Act Tagging for Evaluation (DATE)

The dialogue act labelling of the Communicator corpus follows the DATE tagging scheme (Walker et al., 2001). DATE classifies each utterance along three cross-cutting orthogonal dimensions of utterance classification: (1) a SPEECH ACT dimension; (2) a CONVERSATIONAL-DOMAIN dimension; and (3) a TASK-SUBTASK dimension. The SPEECH ACT and CONVERSATIONAL-DOMAIN dimensions are general across domains, while the TASK-SUBTASK dimension involves a task model that is not only domain specific, but could vary from system to system because some systems might make finer-grained subtask distinctions.

Speech-Act	Example
REQUEST-INFO	*And, what city are you flying to?*
PRESENT-INFO	*The airfare for this trip is 390 dollars.*
OFFER	*Would you like me to hold this option?*
STATUS-REPORT	*Accessing the database; this might take a few seconds.*
EXPLICIT-CONFIRM	*You will depart on September 1st. Is that correct?*
IMPLICIT-CONFIRM	*Leaving from Dallas.*
INSTRUCTION	*Try saying a short sentence.*

Figure 3: Example Speech Act utterances

The SPEECH ACT dimension captures distinctions between communicative goals such as requesting information (REQUEST-INFO), presenting information (PRESENT-INFO) and making offers (OFFER) to act on behalf of the caller. Some examples are in Figure 3.

The CONVERSATIONAL-DOMAIN dimension involves the domain of discourse that an utterance is about. DATE distinguishes three domains within this dimension. Examples of each domain are given in Figure 4. The ABOUT-TASK domain is necessary for evaluating a dialogue system's ability to collaborate with a speaker on achieving the task goal. It supports metrics such as the amount of time/effort the system takes to complete a particular phase of making an airline reservation, and any ancillary hotel/car reservations. The ABOUT-COMMUNICATION domain reflects the system goal of managing the verbal channel of communication and providing evidence of what has been understood. Utterances of this type are frequent in human-computer dialogue, where they are motivated by the need to avoid potentially costly errors arising from imperfect speech recognition. All implicit and explicit confirmations are about communication. The ABOUT-SITUATION-FRAME domain pertains to the goal of managing the user's expectations about how to interact with the system.

Conversational Domain	Example
ABOUT-TASK	*And what time didja wanna leave?*
ABOUT-COMMUNICATION	*Leaving from Miami.*
ABOUT-SITUATION-FRAME	*You may say repeat, help me out, start over, or, that's wrong*

Figure 4: Example utterances distinguished within the Conversational Domain Dimension

The TASK-SUBTASK dimension focuses on specifying which subtask of the travel reservation task the utterance contributes to. This dimension distinguishes among 28 subtasks, some of which can also be grouped at a level below the top level task. The TOP-LEVEL-TRIP task describes the task which contains as its subtasks the ORIGIN, DESTINATION, DATE, TIME, AIRLINE, TRIP-TYPE, RETRIEVAL and ITINERARY tasks. The GROUND task includes both the HOTEL and CAR-RENTAL subtasks. The HOTEL task includes both the HOTEL-NAME and HOTEL-LOCATION subtasks.[1]

4.1. Implementation and Metrics Derivation

To label the system utterances in the 2001 Communicator corpus with the DATE dialogue acts, we first applied the dialogue act tagger that was developed for labelling the 2000 Communicator data (Walker et al., 2001). In this tagger, an utterance or utterance sequence is identified and labelled automatically by reference to a database of utterance patterns that are hand-labelled with DATE tags. The collection and DATE labelling of the utterances for the pattern database was done in cooperation with the site developers. Since the systems use template based generation and have only a limited number of ways of saying the same content, very few utterance patterns needed to be hand-labelled when compared to the actual number of utterances occurring in the corpus. Further abstraction on the patterns was done with a named-entity labeler which replaces specific tokens of city names, airports, hotels, airlines, dates, times, cars, and car rental companies. For example, *what time do you want to leave <AIRPORT> on <DATE-TIME>?* is the typed utterance for *what time do you want to leave Newark International on Monday?*. For the 2000 tagging, the number of utterances in the pattern database was 1700 whereas the total number of utterances in the 662 dialogues was 22930. The named-entity labeller was also applied to the system utterances in the corpus. We collected vocabulary lists from all the sites for the named-entity task. In

[1] Certain utterances in the dialogues are not specific to any particular task and can be used for any subtask, for example, system statements that it misunderstood. These utterances are given a "meta" dialogue act status in the task dimension. There are 13 such dialogue acts distinguished within DATE.

most cases, systems had preclassified the individual tokens into generic types.

The tagger implements a simple pattern matching algorithm to do the dialogue act labelling. Utterance patterns in the pattern database are matched in the corpus and the DATE label of that pattern is assigned to the matching pattern in the corpus. The matching ignores punctuation since systems vary in the way they record punctuation.[2]

Certain utterances have different communicative functions depending on the context in which they occur. For example, phrases like *leaving in the <DATE-TIME>* are implicit confirmations when they constitute an utterance on their own, but are part of the flight information presentation when they occur embedded in utterances such as *I have one flight leaving in the <DATE-TIME>*. To prevent incorrect labelling for such ambiguous cases, the pattern database is sorted so that sub-patterns are matched after the patterns within which they are embedded.

While this tagger was 100% accurate for the 2000 data, when applied to the 2001 data it was able to label only 60% of the data, where the coverage is calculated on the character counts of the utterances. On examination of the unlabelled utterances, we found that many systems had augmented their inventory of vocabulary items as well as utterances for the 2001 data collection. As a result, there were many new patterns unaccounted for in the existing named-entity lists as well as the pattern database. In an attempt to cover the remaining 40% of the data, we therefore augmented the named-entity lists by obtaining a new set of preclassified vocabulary items from the sites, and added 800 hand-labelled patterns to the pattern database. For the labelling of any additional unaccounted for patterns, we implemented a contextual rule-based postprocessor that looks at the surrounding dialogue acts of an unmatched utterance within a turn and attempts to label it. Figure 5 shows the current DATE tagging system. The contextual rules are intended to capture rigid system dialogue behaviors that are reflected in the DATE sequences within a turn.[3] For example, one very frequently occurring DATE sequence within system turns is present_info:flight, present_info:price, offer:flight, and some of the rules use this contextual information to tag unlabelled utterances: if the postprocessor encounters a turn in which the first two utterances have been labelled with present_info:flight and present_info:price, and the third utterance is left unlabeled by the pattern matcher, it uses the above rule to assign the third utterance with the present_info:price label.[4] Not all turn-internal DATE sequences could be used as contextual rules, however, because many of them are highly ambiguous. For example, about_comm:apology:meta_slu_reject can be followed by a system instruction as well as any kind of request for information (typically) repeated from the previous system utterance.

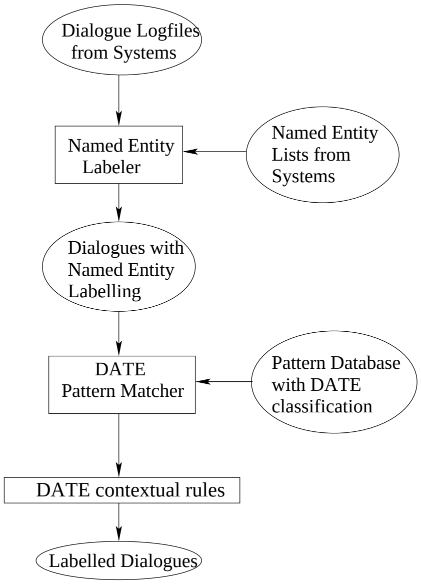

Dialogue Logfiles from Systems
Named Entity Labeler
Named Entity Lists from Systems
Dialogues with Named Entity Labelling
DATE Pattern Matcher
Pattern Database with DATE classification
DATE contextual rules
Labelled Dialogues

Figure 5: The DATE Dialogue Act Tagger

The tagger, augmented with the new named entity lists, the new pattern database, and the postprocessor, covers 98.4% of the 2001 data. A hand evaluation of 10 randomly selected dialogues from each system shows that we achieved a classification accuracy of 96% at the utterance level.

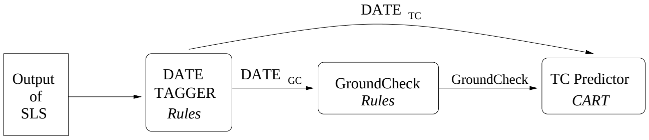
For future evaluations, we would like to avoid the expensive and tedious process that we faced with the 2001 tagging. In (Prasad and Walker, 2002), we have experimented with a machine learning method for DATE tagging. The learner uses a total of 19 features which are either derived directly from the logfiles, derived from the human transcription of the user utterances, or represent aspects of the dialogue context in which the utterance occurs. The dialogue context features include the left unigram and bigram DATE context which extends to the previous system turn, the number of dialogue acts in the turn, the position of the target utterance in the turn, the system utterances to the left and right of the target utterance, and the previous user utterance. The use of all these features is designed to reduce the ambiguity of the dialogue act context. We have trained and tested the automatic DATE tagger on various combinations of the Communicator 2000 and 2001 human-computer corpora. The accuracy of a DATE tagger trained and tested on the 2000 corpus is 98.5%. On the 2001 corpus, it achieves an accuracy of 71.8%, but the accuracy improves to 93.8% when just 3000 utterances from the 2001 corpus are added to the training data (with the test data being the remainder of the 2001 corpus). These results suggest that it is possible to automatically label system utterances for future evaluations without much additional effort.

[2] Ignoring punctuation does not, however, create an utterance segmentation problem for the tagger. The utterances in the pattern database provide the reference points for utterance boundaries.

[3] The logfile standard distinguishes system and user turns within the dialogues.

[4] The DATE labels have three fields separated by ":" corresponding to the three dimensions of the DATE scheme. The first field describes the utterance in the conversational domain dimension, the second in the speech act dimension, and the third in the task-subtask dimension. For the about-task dimensions, only the second and the third fields are given in the labels.

5. Automatically Predicting TaskCompletion using DATE Dialogue Acts



DATE_{TC} = {acknowledgement:flight_booking, request_info:airline, request_info:top_level_trip}

DATE_{GC} = {request_info:hotel_name, request_info:hotel_location, offer:hotel, offer:rental}

Figure 6: Schema for TaskCompletion Prediction

As discussed in Section 3., one of the features used in the Regression Tree for UserSatisfaction prediction is TaskCompletion. In order to make the system completely automatic, an approximation of this feature is derived by training a Classification Tree using DATE dialogue acts.

Figure 6 shows the methodology behind TaskCompletion prediction. A Classification Tree is trained that categorizes dialogues into TaskCompletion=0, TaskCompletion=1 or TaskCompletion=2. The baseline for this experiment is 59.3% which is the number of dialogues where TaskCompletion=1.

The first stage is to infer the DATE dialogue acts - this method is detailed in Section 4.. Each DATE dialogue act is tallied and the counts are used as features to train the CART tree. An additional feature is GroundCheck which is instantiated by looking for DATE labels relating to whether ground arrangements have been made. A simple set of rules searches for one of the following DATE dialogue acts - request_info:hotel_name; request_info:hotel_location; offer:hotel; and offer:hotel_rental. These DATE types are picked because the spoken dialogue systems use these once car or hotel arrangements have been requested by the user. The GroundCheck feature is binary: 0 if none of the above labels are observed and 1 if any or all are observed.

The trained tree classifies dialogues into the three TaskCompletion categories with an accuracy of 85.0%. As illustrated in Figure 6, this tree uses 4 different DATE acts to predict TaskCompletion - GroundCheck, acknowledgment:flight_booking, request_info:airline and request_info:top_level_trip. The structure of the tree is such that GroundCheck divides the data into TaskCompletion <2 and TaskCompletion=2. If GroundCheck=0 and there is an acknowledgment of a booking then we can take it that it is the flight that has been booked and therefore TaskCompletion=1. Interestingly, if there is no acknowledgment of a booking then TaskCompletion=0, unless the system got to the stage of asking the user for airline preference and if request_info:top_level_trip<2. More than one request_info:top_level_trip indicates that there was a problem in the dialogue and a START-OVER occurred.

The tree that predicts binary TaskCompletion has an accuracy of 92.0% with a baseline of 85%. This is a simple tree that checks if an acknowledgment:flight_booking has occurred. If it has, then TaskCompletion=1, otherwise it looks for the DATE act about_situation_frame:instruction:meta_situation_info which captures the fact that the system has told the user what the system can and cannot do or has informed the user about the current state of the task. This must help with TaskCompletion as the tree tells us that if one or more of these acts are observed then TaskCompletion=1, otherwise TaskCompletion=0.

6. Experimental Design for UserSatisfaction Prediction

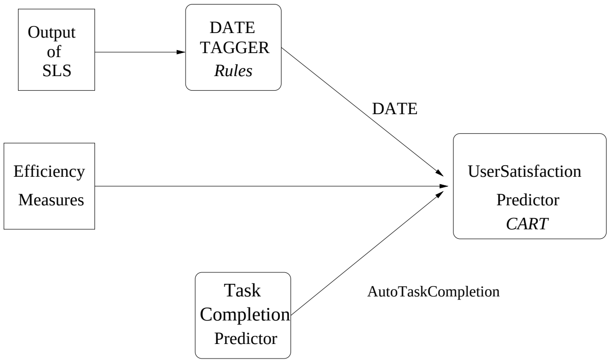


Figure 7: Schema for UserSatisfaction Prediction

To apply CART to the training of the UserSatisfaction predictor, each dialogue is taken as a vector of a set of input features and UserSatisfaction is taken as the response variable. As discussed in the Section 2. and shown in Figure 1, there are three groups of metrics used in the PARADISE framework: Task Success, Efficiency Measures and Qualitative Measures. Similarly, the types of features used to train the Regression Tree fall into the same three categories as illustrated in Figure 7, which shows the system design for automatically predicting UserSatisfaction. A comprehensive list of all the features are given in Table 8.

Firstly, Task Success is captured by the TaskCompletion feature which can be either hand-labelled or automatically predicted by the method described in 5.. The Regression Tree is trained using the hand-labelled TaskCompletion feature. If one was to test this system on new unseen data, one would use the automatically predicted AutoTaskCompletion in place of the hand-labelled TaskCompletion. We present results for testing on both the hand-labelled and automatically obtained TaskCompletion.

Secondly, Efficiency Measures are captured by metrics taken from the logfile. These are divided into 2 sets: hand-labelled and automatically extracted. Word Error Rate (WERR), Sentence Error Rate (SERR) all require a transcription of the words and are, therefore, classified as hand-labelled. TurnsPerTask (number of turns in dialogue), TimeOnTask (in seconds), MeanWrdsPerUsrTurn are all automatically extractable from the logfile. We assume phone-type is automatically derivable by automatic number identification (ANI) and that session number can be extracted from the logfile.

- **Task Success Features**
  - *Hand-labelled*
    - HLTaskCompletion
  - *Automatic*
    - AutoTaskCompletion
- **Efficiency Measures**
  - *Hand-labelled*
    - WERR, SERR
  - *Automatic*
    - TimeOnTask, TurnsOnTask, NumOverlaps, MeanUsrTurnDur, MeanWrdsPerUsrTurn, MeanSysTurnDur, MeanWrdsPerSysTurn, DeadAlive
    - Phone-type, SessionNumber
- **Qualitative Measures**
  - Automatic DATE Unigrams
    - present_info:flight, present_info:price etc..
  - Automatic DATE Bigrams
    - present_info:flight+present_info:price etc..

Figure 8: Features used to train the UserSatisfaction Prediction Tree

Finally, the quality of the dialogue is captured by the different DATE dialogue act frequencies. We found that the distribution of DATE acts were better captured by using the frequency normalized over the total number of dialogue acts. In addition to these unigram proportions, the bigram frequencies of the DATE dialogues acts were also calculated.

7. Results for UserSatisfaction Prediction

The results of the UserSatisfaction prediction Regression Tree are given in terms of the correlation between the predicted UserSatisfaction and actual UserSatisfaction as calculated from the survey. Here, we also quote correlation and R^2 for comparison with previous studies.

Table 1 gives the correlation results for UserSatisfaction prediction using different sets of features and hand-labelled or automatically predicted TaskCompletion. The first column gives the results using only the automatically extracted Efficiency Measures which give a correlation of 0.5955 ($R^2 = .355$) using hand-labelled TaskCompletion. Adding the hand-labelled Efficiency Measures increases this result to 0.607 ($R^2 = .368$). This, however, is not as good as just using the automatic Efficiency Measures and the DATE features in combination with TaskCompletion which yields a correlation of 0.614 ($R^2 = .377$). This result is the same as using all the groups of features as given in the final column. As the DATE features are chosen over the hand-labelled Efficiency Measures, this shows that they are more discriminatory in determining UserSatisfaction.

The discriminatory use of the DATE features is seen more when used in conjunction with the automatic TaskCompletion. Here, we see an increase from 0.4593 ($R^2 = .21$) to 0.484 ($R^2 = .234$) when the DATE features are added to the automatic Efficiency and TaskCompletion features. This is likely due to the fact that the DATE features compensate for the inaccuracies of the automatic TaskCompletion by marking landmarks in the dialogue where parts of the task have been completed, such as about_communication:implicit_confirm:depart_arrive_time or request_info:price, as illustrated in the Regression Tree given in 9, 10 and 11. This tree is formed using the automatic Efficiency and DATE features which has a correlation of 0.614/0.484. The interpretation of the tree is given in the following section.

7.1. Regression Tree Interpretation

Diagrams of the trained decision trees are given below. At any junction, if the query is true then one takes the path down the right-hand side of the tree, otherwise one takes the left-hand side. The leaf nodes contain the predicted value.

Figure 9 illustrates that TaskCompletion is at the top of the tree and is, therefore, the most queried feature. The phone-type is an important part of UserSatisfaction prediction, whereby dialogues conducted over corded phones have higher satisfaction. This is likely to be due to better recognition performance from corded phones. The rule containing the bigram request_info:depart_arrive_date+USER states that if there is more than one occurrence of this request then UserSatisfaction will be lower. A repetition of this DATE act indicates that a misunderstanding occurred the first time it is requested or that the task is multi-leg in which case UserSatisfaction is generally lower.

Figure 10 gives part of the left side of the tree where TaskCompletion >0 i.e. some level of TaskCompletion has been achieved. This portion of the tree shows how important dialogue length is to UserSatisfaction. TurnsOnTask is the number of turns which are task-oriented, for example, initial instructions on how to use the system are not taken as a TurnOnTask. The tree indicates that dialogues which are long (TurnsOnTask $>$ 110) are satisfactory (UserSatisfaction = 15.2) if some task is completed (TC >0). Again, if the phone-type is not corded UserSatisfaction is lower.

Figure 11 gives the final, lower part of the tree. If there has been more than three acknowledgments of bookings, this indicates that several legs of a journey have been successfully booked and, therefore, UserSatisfaction is high, in particular if the system has asked if the user would like a price for their itinerary. This request is one of the final dialogue acts a system does before the task is completed.

The DATE act about_comm:apology:meta_slu_reject is a measure of the system's level of misunderstanding. Therefore, the more of these dialogue act types the lower UserSatisfaction. This part of the tree uses length in a similar way described earlier, whereby long dialogues are only allocated lower UserSatisfaction if they do not involve ground arrangements. In longer dialogues, users seem to prefer systems that include a number of implicit confirmations as these dialogues have higher UserSatisfaction.

The dialogue act request_info:top_level_trip usually occurs at the start of the dialogue and requests the initial travel plan. If there are more than one of this request_trip dialogue act, it indicates that a START-OVER occurred due to system failure, this leads to lower UserSatisfaction.

Feature used	**Auto Eff.**	**Auto Eff. +HL Eff.**	**Auto Eff. + DATE**	**Auto Eff. + HL Eff. + DATE**
HLTaskCompletion	0.5955	0.607	0.614	0.614
AutoTaskCompletion	0.4593	0.476	0.484	0.484

Table 1: Correlation results using Automatic Efficiency Measures (Eff.), adding DATE features and hand-labelled Efficiency Measures, for trees tested on either hand-labelled or automatically derived TaskCompletion

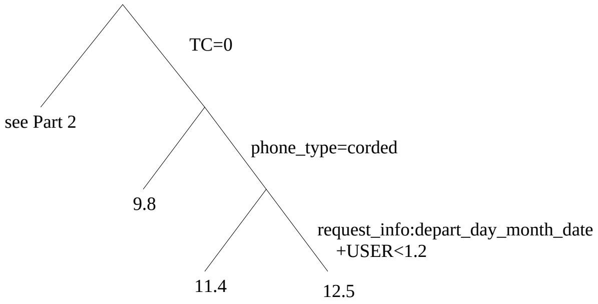


Figure 9: Sub-tree (part 1) of the Regression Tree for UserSatisfaction (TC is binary TaskCompletion)

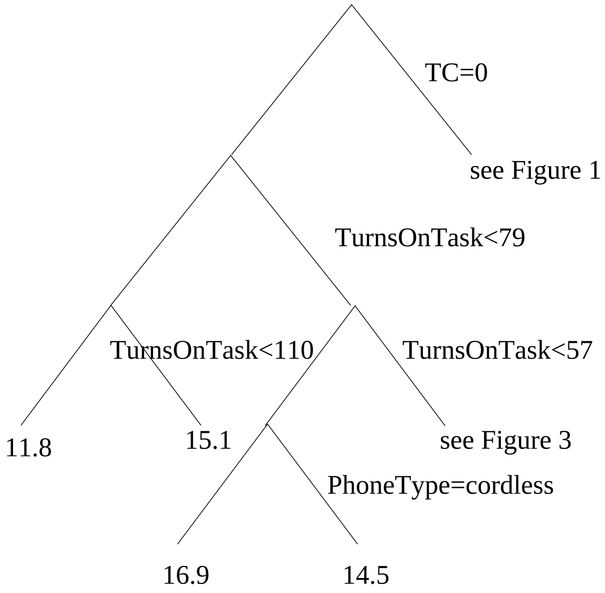


Figure 10: Sub-tree (part 2) of the Regression Tree for UserSatisfaction (TC is binary TaskCompletion)

A figure known as Feature Usage Frequency can be calculated from the CART tree. This metric reflects which features are the most discriminatory in the tree. Specifically, this measure is the number of times a feature is queried during the regression calculation of each data point. The figure is normalized so that the feature usage sums to one for each tree. It reflects the position in the Regression Tree as the higher the feature is in the tree, the more times it is queried. Efficiency Measures are the most discriminatory feature set covering 47% of the queries. The Dialogue Act Quality Measures account for 32% of the tree's discriminatory features. Task Success is the feature queried at the top of the tree and accounts for 21% of the feature usage.

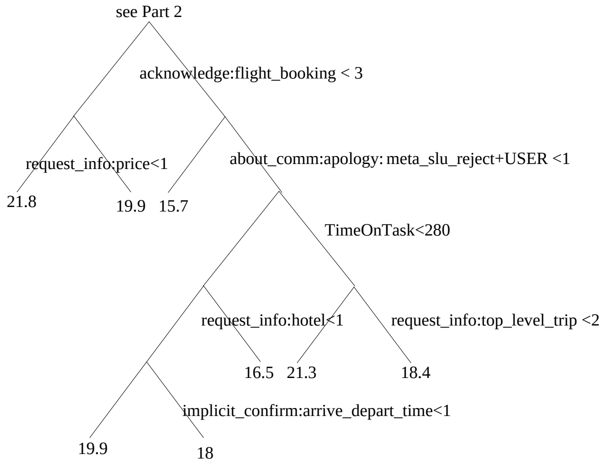

see Part 2
acknowledge:flight_booking < 3
request_info:price<1
about_comm:apology: meta_slu_reject+USER <1
21.8
19.9
15.7
TimeOnTask<280
request_info:hotel<1
request_info:top_level_trip <2
16.5
21.3
18.4
implicit_confirm:arrive_depart_time<1
19.9
18

Figure 11: Sub-tree (part 3) of the Regression Tree for UserSatisfaction

8. Previous Work

Previous work looks at predicting UserSatisfaction using multi-variate linear regression using non-automatic efficiency, quality and task success metrics (Walker et al., 2000; Walker et al., 2001). This work looks at predicting UserSatisfaction for the 2000 Communicator data and finds that adding the counts for the DATE to the logfile standard metrics yields an increase from .37 to 0.42 (R^2).

These results are not directly comparable as they are performed on different data. However, a qualitative comparison is interesting. Their multi-variate linear regression experiments give coefficients for each feature which indicate the magnitude and whether the metric is a positive or negative predictor of UserSatisfaction. Some of the metrics which are heavily weighted are comparable to the ones the Regression Tree finds. For example, TaskCompletion has the highest magnitude coefficient and comes at the top of our regression tree. Task duration is negatively weighted. This is also the case in our Regression Tree, although the decision tree also captures non-linear interactions between features. For example, longer dialogues are only penalized if a more simple task is being performed.

DATE acts used in both systems include acknowledgment:flight_booking, request_info:depart_arrive_date and request_info:top_level_trip. Explicit confirmations have a higher weighting than implicit confirmations in the linear-regression model, whereas our CART tree tends to favor implicit confirmations. This may indicate that the spoken dialogue systems are tending towards more natural conversations where implicit confirmations are preferred.

Another area of related work is that on detecting “Problematic dialogues" (Walker et al., 2002). The goal of these Problematic Dialogue Predictors is to determine which dialogues are likely to fail before the end of the dialogue so that the system can be adapted on-line, and the user can be transferred to a human customer care agent if there is a problem. This is similar to our TaskCompletion predictor in the Communicator domain. They use features from a number of sources, such as acoustic features and features from the natural language understanding and dialogue manager components. The most important features come from the Natural Language Understanding system (such as interpretation confidence measures). For the Communicator Data, such detailed metrics are not available for interpretation, although the DATE act about_comm:apology:meta_slu_reject does approximate this. In our study, predicting TaskCompletion prior to dialogue completion is also a possibility. However, more sophisticated features (such as ASR and NLU confidence scores) would need to be used in order for this to be a possibility.

9. Conclusion

In summary, we have presented results for automatically evaluating system performance in the October-2001 corpus of 1242 Communicator dialogues in the travel planning domain. In this study, performance is measured either with a Task Success metric or with UserSatisfaction. As predictors of UserSatisfaction, we examined the utility of three different types of features: Task Success Features, Efficiency Measures and Dialogue Act Quality metrics. We described how we automatically labelled the dialogues in order to create the dialogue act features that were input for our automatic prediction task. We showed that using these features, we can predict TaskCompletion with an accuracy of 85% and model UserSatisfaction with a correlation of up to 0.614.

A possible extension to this work is the automatic prediction of the *user's* dialogue act type. The user's utterance would be predicted using a set of dialogue act type specific language models run over the ASR output and a dialogue model (Wright, 2000). This dialogue model would be highly predictive, given that we have an accurate DATE tagger for the system's turns combined with the fact that users do not take the initiative frequently in the Communicator dialogues. A further extension of this work is to look at the intonation of the user. For example, if the system is expecting a short yes/no answer and the user replies with a long utterance with a rising intonation contour then this is more likely to be a question, indicating a breakdown in the initial dialogue strategy.

10. Acknowledgments

The work reported in this paper was partially funded by DARPA contract MDA972-99-3-0003.

11. References

Bonneau-Maynard, H., L. Devillers, and S. Rosset, 2000. Predictive performance of dialog systems. In *Language Resources and Evaluation Conference*.

Brieman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone, 1984. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey California.

Cattoni, Roldana, Morena Danieli, Andrea Panizza, V. Sandrini, and C. Soria, 2001. Building a corpus of annotated dialogues: the ADAM experience. In *Proc. of the Conference Corpus-Linguistics-2001, Lancaster, U.K.*.

Doyle, Jon, 1992. Rationality and its roles in reasoning. *Computational Intelligence*, 8(2):376–409.

Kamm, Candace, 1995. User interfaces for voice applications. In David Roe and Jay Wilpon (eds.), *Voice Communication between Humans and Machines*. National Academy Press, pages 422–442.

Keeney, R. and H. Raiffa, 1976. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons.

Lamel, Lori and Sophie Rosset, 2000. Considerations in the design and evaluation of spoken language dialog systems. In *ISCLP*.

Polifroni, Joseph, Lynette Hirschman, Stephanie Seneff, and Victor Zue, 1992. Experiments in evaluating interactive spoken language systems. In *Proceedings of the DARPA Speech and NL Workshop*.

Prasad, Rashmi and M. Walker, 2002. Training a dialogue act tagger for human-human and human-computer travel dialogues. In *submission to ACL*.

Shriberg, Elizabeth, Elizabeth Wade, and Patti Price, 1992. Human-machine problem solving using spoken language systems (SLS): Factors affecting performance and user satisfaction. In *Proceedings of the DARPA Speech and NL Workshop*.

Walker, M., I. Langkilde-Geary, H. Wright Hastie, J. Wright, and A. Gorin, 2002. Automatically training a problematic dialogue predictor for a spoken dialogue system. *JAIR*.

Walker, M., R. Passonneau, and J. Boland, 2001. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *Proceedings of the 39rd Annual Meeting of the Association for Computational Linguistics (ACL/EACL-2001)*.

Walker, M. A., 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12:387–416.

Walker, M. A., C. A. Kamm, and D. J. Litman, 2000. Towards developing general models of usability with PARADISE. In *Natural Language Engineering: Special Issue on Best Practice in Spoken Dialogue Systems*.

Wright, H., 2000. *Modelling Prosodic and Dialogue Information for Automatic Speech Recognition*. Ph.D. thesis, University of Edinburgh.