Evaluating Interactive Dialogue Systems: Extending Component Evaluation to Integrated System Evaluation

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm and Alicia Abella
AT&T Labs—Research
180 Park Avenue
Florham Park, NJ 07932-0971 USA
walker,diane,cak,abella@research.att.com

Abstract

This paper discusses the range of ways in which spoken dialogue system components have been evaluated and discusses approaches to evaluation that attempt to integrate component evaluation into an overall view of system performance. We will argue that the PARADISE (PARAdigm for DIalogue System Evaluation) framework has several advantages over other proposals.

1 Introduction

Interactive spoken dialogue systems are based on many component technologies: speech recognition, text-to-speech, natural language understanding, natural language generation, and database query languages. While evaluation metrics for these components are well understand (Sparck-Jones and Galliers, 1996; Walker, 1989; Hirschman et al., 1990), it has been difficult to develop standard metrics for complete systems that integrate all these technologies. One problem is that there are so many potential metrics that can be used to evaluate a dialog system. For example, a dialog system can be evaluated by measuring the system's ability to help users achieve their goals, the system's robustness in detecting and recovering from errors of speech recognition or of understanding, and the overall quality of the system's interactions with users (Danieli and Gerbino, 1995; Hirschman and Pao, 1993; Polifroni et al., 1992; Price et al., 1992; Simpson and Fraser, 1993). Another problem is that dialog evaluation is not reducible to transcript evaluation, or to comparison with a wizard's reference answers (Bates and Ayuso, 1993; Polifroni et al., 1992; Price et al., 1992), because the set of potentially acceptable dialogs can be very large.

Current proposals for dialog evaluation metrics are both *objective* and *subjective*. The objective metrics that have been used to evaluate a dialog as a whole include (Abella, Brown, and Buntschuh, 1996; Ciaremella, 1993; Danieli and Gerbino, 1995; Hirschman et al., 1990; Hirschman et al., 1993; Polifroni et al., 1992; Price et al., 1992; Smith and Hipp, 1994; Smith and Gordon, 1997; Walker, 1996):

- percentage of correct answers with respect to a set of reference answers
- transaction success, task completion, or quality of solution
- number of turns or utterances;
- dialogue time or task completion time
- mean user response time
- mean system response time
- frequency of diagnostic error messages
- percentage of "non-trivial" (more than one word) utterances.
- mean length of "non-trivial" utterances

Objective metrics can be calculated without recourse to human judgement, and in many cases, can be calculated automatically by the spoken dialogue system. One possible exception is task-based success measures, such as transaction success, task completion or quality of solution metrics, which can be either an objective or a subjective measure depending on whether the users' goals are well-defined at the beginning of the dialogue. This is the case in controlled experiments, but in field studies, determining whether the user accomplished the task requires subjective judgements.

Subjective metrics require subjects using the system or human evaluators to categorize the dialogue or utterances within the dialog along various qualitative dimensions. Because these metrics are based on human judgements, such judgements need to be reliable across judges in order to compete with the reproducibility of metrics based on objective criteria. Subjective metrics can still be quantitative, as when a ratio between two subjective categories is computed. Subjective metrics that have been used include (Danieli and Gerbino, 1995; Hirschman and Pao, 1993; Simpson and Fraser, 1993; Danieli et al., 1992; Bernsen, Dybkjaer, and Dybkjaer, 1996) :

- *Implicit recovery (IR)*: the system's ability to use dialog context to recover from errors of partial recognition or understanding.
- *Explicit Recovery*: the proportion of explicit recovery utterances made by both the system *system turn*

correction (STC), and the user, *user turn correction (UTC)*.

- *Contextual appropriateness (CA)*: the coherence of system utterances with respect to dialog context. Utterances can be either *appropriate (AP)*, *inappropriate (IP)*, or *ambiguous (AM)*.
- Cooperativity of system utterances: classified on the basis of the adherance of the system's behavior to Grice's conversational maxims (Grice, 1967).
- Correct and Partially Correct Answers.
- Appropriate or Inappropriate Directives and Diagnostics: directives are instructions the system gives to the user, while diagnostics are messages in which the system tells the user what caused an error or why it can't do what the user asked.
- User Satisfaction: a metric that attempts to captures user's perceptions about the usability of the system. This is usually assessed with multiple choice questionnaires that ask users to rank the system's performance on a range of usability features according to a scale of potential assessments.

Both the objective and the subjective metrics have been very useful to the spoken dialogue community in comparing different systems for carrying out the same task, but these metrics are also limited.

One widely acknowledged limitation is that the use of reference answers makes it impossible to compare systems that use different dialog strategies for carrying out the same task. The reference answer approach requires canonical responses (i.e., a single "correct" answer) to be defined for every user utterance. Thus it is not possible to use the same reference set to evaluate a system that may choose to give a summary as a response in one case, ask a disambiguating question in another, or respond with a set of database values in another.

A second limitation is that various metrics may be highly correlated with one another, and provide redundant information on performance. Determining correlations requires a suite of metrics that are widely used, and testing whether correlations hold across multiple dialogue applications.

A third limitation arises from the inability to tradeoff or combine various metrics and to make generalizations (Fraser, 1995; Sparck-Jones and Galliers, 1996). For example, consider a comparison of two train timetable information agents (Danieli and Gerbino, 1995), where Agent A in Dialogue 1 uses an explicit confirmation strategy, while Agent B in Dialogue 2 uses an implicit confirmation strategy:

(1) User: I want to go from Torino to Milano.
Agent A: Do you want to go from Trento to Milano? Yes or No?
User: No.

(2) User: I want to travel from Torino to Milano.
Agent B: At which time do you want to leave from Merano to Milano?
User: No, I want to leave from Torino in the evening.

Danieli and Gerbino found that Agent A had a higher transaction success rate and produced less inappropriate and repair utterances than Agent B. In addition, they found that Agent A's dialogue strategy produced dialogues that were approximately twice as long as Agent B's, but they could not determine whether Agent A's higher transaction success or Agent B's efficiency was more critical to performance.

The ability to identify factors that affect performance is a critical basis for making generalizations across systems performing different tasks (Cohen, 1995; Sparck-Jones and Galliers, 1996). It would be useful to know how users' perceptions of performance depend on the strategy used, and on tradeoffs among factors such as efficiency, speed, and accuracy. In addition to agent factors such as the differences in dialogue strategy seen in Dialogues 1 and 2, task factors such as database size and environmental factors such as background noise may also be relevant predictors of performance.

In the remainder of this paper, we discuss the PARADISE framework (PARAdigm for DIalogue System Evaluation) (Walker et al., 1997), and that it addresses these limitations, as well as others. We will show that PARADISE provides a useful methodology for evaluating dialog systems that integrates and enhances previous work.

2 Integrating Previous Approaches to Evaluation in the PARADISE Framework

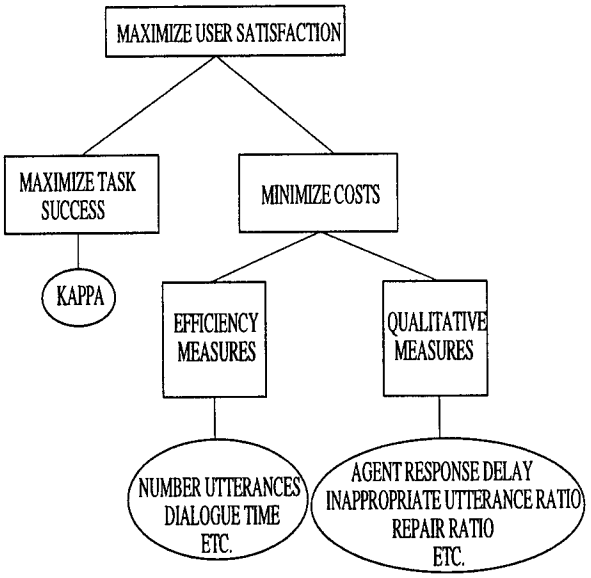

Figure 1: PARADISE's structure of objectives for spoken dialogue performance

The PARADISE framework for spoken dialogue evaluation is based on methods from decision theory (Keeney and Raiffa, 1976; Doyle, 1992), which supports combin-

ing the disparate set of performance measures discussed above into a single performance evaluation function. 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 (rectangles) shown in Figure 1.

At the top level, this model 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. User satisfaction ratings (Kamm, 1995; Shriberg, Wade, and Price, 1992; Polifroni et al., 1992) are the most widely used external indicator of the usability of a dialogue agent.

The model further posits that two types of factors are potential relevant contributors to user satisfaction, namely task success and dialogue costs. PARADISE uses linear regression to quantify the relative contribution of the success and cost factors to user satisfaction. The task success measure builds on previous measures of transaction success and task completion (Danieli and Gerbino, 1995; Polifroni et al., 1992), but makes use of the Kappa coefficient (Carletta, 1996; Siegel and Castellan, 1988) to operationalize task success.

The cost factors consist of two types. The efficiency measures arise from the list of objective performance measures used in previous work as described above. Qualitative measures try to capture aspects of the quality of the dialog. These are based on both objective and subjective measures used in previous work, such as the frequency of diagnostic or error messages, inappropriate utterance ratios, or the proportion of repair utterances.

The remainder of this section explains the measures (ovals in Figure 1) used to operationalize the set of objectives, and the methodology for estimating a quantitative performance function that reflects the objective structure. Section 2.1 describes PARADISE's task representation, which is needed to calculate the task-based success measure described in Section 2.2. Section 2.3 describes the cost measures considered in PARADISE, which reflect both the efficiency and the naturalness of an agent's dialogue behaviors. Section 2.4 describes the use of linear regression and user satisfaction to estimate the relative contribution of the success and cost measures in a single performance function. Finally, Section 2.5 summarizes the method.

2.1 Tasks as Attribute Value Matrices

A general evaluation framework requires a task representation that decouples *what* an agent and user accomplish from *how* the task is accomplished using dialogue strategies. PARADISE uses an *attribute value matrix (AVM)* to represent dialogue tasks. The AVM consists of the information that must be exchanged between the agent and the user during the dialogue, represented as a set of ordered pairs of attributes and their possible values.[1]

As a first illustrative example, consider a simplification of the train timetable domain of Dialogues 1 and 2, where the timetable only contains information about rush-hour trains between four cities, as shown in Table 1. This AVM consists of four attributes (abbreviations for each attribute name are also shown).[2] In Table 1, these attribute-value pairs are annotated with the direction of information flow to represent who acquires the information, although this information is not used for evaluation. During the dialogue the agent must acquire from the user the values of DC, AC, and DR, while the user must acquire DT.

attribute	possible values	information flow
depart-city (DC)	Milano, Roma, Torino, Trento	to agent
arrival-city (AC)	Milano, Roma, Torino, Trento	to agent
depart-range (DR)	morning,evening	to agent
depart-time (DT)	6am,8am,6pm,8pm	to user

Table 1: Attribute value matrix, simplified train timetable domain

A1:	Hello, This is Train Enquiry Service.	DC,AC,DR,DT
	Please speak after the tone.	DC,AC,DR,DT
	Which information do you need?	DC,AC,DR,DT
U1:	I want to go from Torino to Milano.	DC,AC
A2:	Do you want to go from Trento to Milano?	DC,AC
	Yes or No?	DC,AC
U2:	No.	DC,AC
A3:	Do you want to leave from Trento?	DC
	Yes or No?	DC
U3:	No.	DC
A4:	Where do you want to leave from?	DC
U4:	I want to leave from Torino.	DC
A5:	Do you want to leave from Torino?	DC
	Yes or No?	DC
U5:	Yes.	DC
A6:	Do you want to go to Milano?	AC
U6:	Yes.	AC
A7:	At which time do you want to leave?	DR
U7:	I want to travel in the evening.	DR
A8:	Do you want to leave between 6 and 9 p.m.?	DR
	Yes or No?	DR
U8:	Yes.	DR
A9:	There is a train leaving at 8:00 p.m.	DT
	...	

Figure 2: Agent A dialogue interaction (Danieli and Gerbino, 1995)

Like previous approaches to evaluation, performance evaluation using PARADISE requires a corpus of dialogues between users and the agent, in which users execute a set of scenarios. Each scenario execution has a corresponding AVM instantiation indicating the task information requirements for the scenario, where each attribute is paired with the attribute value obtained via the dialogue.

[1] For infinite sets of values, actual values found in the experimental data constitute the required finite set.

[2] The AVM serves as an evaluation mechanism only. We are not claiming that AVMs determine an agent's behavior or serve as an utterance's semantic representation.

B1:	Hello, This is Train Enquiry Service.	DC,AC,DR,DT
	Please speak after the tone.	DC,AC,DR,DT
	Which information do you need?	DC,AC,DR,DT
U1:	I want to travel from Torino to Milano.	DC,AC
B2:	At which time do you want to leave from Merano to Milano?	DC,AC,DR
U2:	No, I want to leave from Torino in the evening.	DC, DR
B3:	Do you want to leave from Torino between 6 and 11 p.m?	DC, DR
	Please answer Yes or No.	DC, DR
U3:	Yes.	DC, DR
B4:	A train leaves at 8 p.m.	DT
	...	

Figure 3: Agent B dialogue interaction (Danieli and Gerbino, 1995)

attribute	actual value
depart-city	Torino
arrival-city	Milano
depart-range	evening
depart-time	8pm

Table 2: Attribute value matrix instantiation, scenario key for Dialogues 1 and 2

For example, assume that a scenario requires the user to find a train from Torino to Milano that leaves in the evening, as in the longer versions of Dialogues 1 and 2 in Figures 2 and 3.[3] Table 2 contains an AVM corresponding to a "key" for this scenario. All dialogues resulting from execution of this scenario in which the agent and the user correctly convey all attribute values (as in Figures 2 and 3) would have the same AVM as the scenario key in Table 2. The AVMs of the remaining dialogues would differ from the key by at least one value. Thus, even though the dialogue strategies in Figures 2 and 3 are radically different, the AVM task representation for these dialogues is identical and the performance of the system for the same task can thus be assessed on the basis of the AVM representation.

2.2 Measuring Task Success

Success at the task for a whole dialogue (or subdialogue) is measured by how well the agent and user achieve the information requirements of the task by the end of the dialogue (or subdialogue). This section explains how PARADISE uses the Kappa coefficient (Carletta, 1996; Siegel and Castellan, 1988) to operationalize the task-based success measure in Figure 1.

The Kappa coefficient, κ, is calculated from a confusion matrix that summarizes how well an agent achieves the information requirements of a particular task for a set of dialogues instantiating a set of scenarios.[4] For example, Table 3 shows a hypothetical confusion matrix that could have been generated in an evaluation of 100 complete dialogues with train timetable agent A (perhaps using the confirmation strategy illustrated in Figure 2).[5] When comparing Agent A to Agent B, a similar table would also be constructed for Agent B.

In Table 3, the values in the matrix cells are based on comparisons between the dialogue and scenario key AVMs. Table 3 summarizes how the 100 AVMs representing each dialogue with Agent A compare with the AVMs representing the relevant scenario keys. Labels v1 to v4 in each matrix represent the possible values of depart-city shown in Table 1; v5 to v8 are for arrival-city, etc. Columns represent the key, specifying which information values the agent and user were supposed to communicate to one another given a particular scenario. Rows represent the data collected from the dialogue corpus, reflecting what attribute values were actually communicated between the agent and the user.

Whenever an attribute value in a dialogue (i.e., data) AVM *matches* the value in its scenario key, the number in the appropriate diagonal cell of the matrix (boldface for clarity) is incremented by 1. The off diagonal cells represent *misunderstandings* that are not corrected in the dialogue. Note that depending on the strategy that a spoken dialogue agent uses, confusions across attributes are possible, e.g., "Milano " could be confused with "morning." The effect of misunderstandings that *are* corrected during the course of the dialogue are reflected in the costs associated with the dialogue, as will be discussed below.

Given a confusion matrix M, success at achieving the information requirements of the task is measured with the Kappa coefficient (Carletta, 1996; Siegel and Castellan, 1988):

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

P(A) is the proportion of times that the AVMs for the actual set of dialogues agree with the AVMs for the scenario keys, and P(E) is the proportion of times that the AVMs for the dialogues and the keys are expected to agree by chance.[6] When there is no agreement other than that which would be expected by chance, $\kappa = 0$. When there is total agreement, $\kappa = 1$. κ is superior to other measures of success such as transaction success (Danieli and Gerbino, 1995), concept accuracy (Simpson and Fraser, 1993), and percent agreement (Carletta, 1996) because κ takes into account the inherent complexity of the task by correcting for chance expected agreement. Thus κ provides a basis for comparisons across agents that are performing *different* tasks.

[3]These dialogues have been slightly modified from (Danieli and Gerbino, 1995). The attribute names at the end of each utterance will be explained below.

[4]Confusion matrices can be constructed to summarize the result of dialogues for any subset of the scenarios, attributes, users or dialogues.

[5]The distributions in the table are roughly based on performance results in (Danieli and Gerbino, 1995).

[6]κ has been used to measure pairwise agreement among coders making category judgments (Carletta, 1996; Krippendorf, 1980; Siegel and Castellan, 1988). Thus, the observed user/agent interactions are modeled as a coder, and the ideal interactions as an expert coder.

	KEY													
	DEPART-CITY				ARRIVAL-CITY				DEPART-RANGE		DEPART-TIME			
DATA	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14
v1	**22**		1		3									
v2		**29**												
v3	4		**16**	4			1							
v4	1	1	5	**11**			1							
v5	3				**20**									
v6						**22**								
v7			2		1	1	**20**	5						
v8			1		1	2	8	**15**						
v9									**45**	10				
v10									5	**40**				
v11											**20**		2	
v12											1	**19**	2	4
v13											2		**18**	
v14											2	6	3	**21**
sum	30	30	25	15	25	25	30	20	50	50	25	25	25	25

Table 3: Confusion matrix, Agent A

When the prior distribution of the categories is unknown, P(E), the expected chance agreement between the data and the key, can be estimated from the distribution of the values in the keys. This can be calculated from confusion matrix M, since the columns represent the values in the keys. In particular:

$$P(E) = \sum_{i=1}^{n} (\frac{t_i}{T})^2$$

where t_i is the sum of the frequencies in column i of M, and T is the sum of the frequencies in M ($t_1 + \ldots + t_n$).

P(A), the actual agreement between the data and the key, is always computed from the confusion matrix M:

$$P(A) = \frac{\sum_{i=1}^{n} M(i,i)}{T}$$

Given the confusion matrix in Table 3, P(E) = 0.079, P(A) = 0.795 and κ = 0.777. Given similar calculations on a confusion matrix for Agent B, we can determine whether Agent A or Agent B is more successful at achieving the task goals.

2.3 Measuring Dialogue Costs

As shown in Figure 1, performance is also a function of a combination of cost measures. Intuitively, cost measures should be calculated on the basis of any user or agent dialogue behaviors that should be minimized. PARADISE supports the use of any of the wide range of cost measures used in previous work, and provides a way of combining these measures by normalizing them.

Each cost measure is represented as a function c_i that can be applied to any (sub)dialogue. First, consider the simplest case of calculating efficiency measures over a whole dialogue. For example, let c_1 be the total number of utterances. For the whole dialogue D1 in Figure 2, c_1(D1) is 23 utterances. For the whole dialogue D2 in Figure 3, c_1(D2) is 10 utterances.

To calculate costs over subdialogues and for some of the qualitative measures, it is necessary to be able to specify which information goals each utterance contributes to. PARADISE uses its AVM representation to link the information goals of the task to any arbitrary dialogue behavior, by tagging the dialogue with the attributes for the task.[7] This makes it possible to evaluate any potential dialogue strategies for achieving the task, as well as to evaluate dialogue strategies that operate at the level of dialogue subtasks (subdialogues).

Consider the longer versions of Dialogues 1 and 2 in Figures 2 and 3. Each utterance in Figures 2 and 3 has been tagged using one or more of the attribute abbreviations in Table 1, according to the subtask(s) the utterance contributes to. As a convention of this type of tagging, utterances that contribute to the success of the whole dialogue, such as greetings, are tagged with all the attributes. Thus the goal of the tagging is to show how the structure of the dialogue reflects the structure of the task (Carberry, 1989; Grosz and Sidner, 1986; Litman and Allen, 1990).

Tagging by AVM attributes is required to calculate costs over subdialogues, since for any subdialogue, task attributes define the subdialogue. For example, the subdialogue about the attribute arrival-city (SA) consists of utterances A6 and U6, its cost c_1(SA) is 2.

Tagging by AVM attributes is also required to calculate the cost of some of the qualitative measures, such as number of repair utterances. (Note that to calculate such costs, each utterance in the corpus of dialogues must also be tagged with respect to the qualitative phenomenon in question, e.g. whether the utterance is a repair.[8]) For example, let c_2 be the number of repair utterances. The repair utterances in Figure 2 are A3 through U6, thus c_2(D1) is 10 utterances and c_2(SA) is 2 utterances. The repair utterance in Figure 3 is U2, but note that according to the AVM task tagging, U2 simultaneously addresses the information goals for arrival-city and depart-range. In

[7] This tagging can be hand generated, or system generated and hand corrected. Preliminary studies indicate that reliability for human tagging is higher for AVM attribute tagging than for other types of discourse segment tagging (Passonneau and Litman, 1997; Hirschberg and Nakatani, 1996).

[8] Previous work has shown that this can be done with high reliability (Hirschman and Pao, 1993).

general, if an utterance U contributes to the information goals of N different attributes, each attribute accounts for 1/N of any costs derivable from U. Thus, c_2(D2) is .5.

Given a set of c_i, it is necessary to combine the different cost measures in order to determine their relative contribution to performance. The next section explains how to combine κ with a set of c_i to yield an overall performance measure.

2.4 Estimating a Performance Function

Given the definition of success and costs above and the model in Figure 1, performance for any (sub)dialogue D is defined as follows:[9]

$$\text{Performance} = (\alpha * \mathcal{N}(\kappa)) - \sum_{i=1}^{n} w_i * \mathcal{N}(c_i)$$

Here α is a weight on κ, the cost functions c_i are weighted by w_i, and $\mathcal{N}$ is a Z score normalization function (Cohen, 1995).

The normalization function is used to overcome the problem that the values of c_i are not on the same scale as κ, and that the cost measures c_i may also be calculated over widely varying scales (e.g. response delay could be measured using seconds while, in the example, costs were calculated in terms of number of utterances). This problem is easily solved by normalizing each factor x to its Z score:

$$\mathcal{N}(x) = \frac{x - \overline{x}}{\sigma_x}$$

where σ_x is the standard deviation for x.

To illustrate the method for estimating a performance function, we will use a subset of the data from Table 3, and add data for Agent B, as shown in Table 4. Table 4 represents the results from a hypothetical experiment in which eight users were randomly assigned to communicate with Agent A and eight users were randomly assigned to communicate with Agent B. Table 4 shows user satisfaction (US) ratings (discussed below), κ, number of utterances (#utt) and number of repair utterances (#rep) for each of these users. Users 5 and 11 correspond to the dialogues in Figures 2 and 3 respectively. To normalize c_1 for user 5, we determine that $\overline{c_1}$ is 38.6 and σ_{c_1} is 18.9. Thus, $\mathcal{N}(c_1)$ is -0.83. Similarly $\mathcal{N}(c_1)$ for user 11 is -1.51.

To estimate the performance function, the weights α and w_i must be solved for. Recall that the claim implicit in Figure 1 was that the relative contribution of task success and dialogue costs to performance should be calculated by considering their contribution to user satisfaction. User satisfaction is typically calculated with surveys that ask users to specify the degree to which they agree with one or more statements about the behavior or the performance of the system. A single user satisfaction measure can be calculated from a single question, or as the mean of a set of ratings. The hypothetical user satisfaction ratings shown in Table 4 range from a high of 6 to a low of 1.

user	agent	US	κ	c_1 (#utt)	c_2 (#rep)
1	A	1	1	46	30
2	A	2	1	50	30
3	A	2	1	52	30
4	A	3	1	40	20
5	A	4	1	23	10
6	A	2	1	50	36
7	A	1	0.46	75	30
8	A	1	0.19	60	30
9	B	6	1	8	0
10	B	5	1	15	1
11	B	6	1	10	0.5
12	B	5	1	20	3
13	B	1	0.19	45	18
14	B	1	0.46	50	22
15	B	2	0.19	34	18
16	B	2	0.46	40	18
Mean(A)	A	2	0.83	49.5	27
Mean(B)	B	3.5	0.66	27.8	10.1
Mean	NA	2.75	0.75	38.6	18.5

Table 4: Hypothetical performance data from users of Agents A and B

Given a set of dialogues for which user satisfaction (US), κ and the set of c_i have been collected experimentally, the weights α and w_i can be solved for using multiple linear regression. Multiple linear regression produces a set of coefficients (weights) describing the relative contribution of each predictor factor in accounting for the variance in a predicted factor. In this case, on the basis of the model in Figure 1, US is treated as the predicted factor. Normalization of the predictor factors (κ and c_i) to their Z scores guarantees that the relative magnitude of the coefficients directly indicates the relative contribution of each factor. Regression on the Table 4 data for both sets of users tests which factors κ, #utt, #rep most strongly predicts US.

In this illustrative example, the results of the regression with all factors included shows that only κ and #rep are significant ($p < .02$). In order to develop a performance function estimate that includes only significant factors and eliminates redundancies, a second regression including only significant factors must then be done. In this case, a second regression yields the predictive equation:

$$\text{Performance} = .40\mathcal{N}(\kappa) - .78\mathcal{N}(c_2)$$

i.e., α is .40 and w_2 is .78. The results also show κ is significant at $p < .0003$, #rep significant at $p < .0001$, and the combination of κ and #rep account for 92% of the variance in US, the external validation criterion. The factor #utt was not a significant predictor of performance, in part because #utt and #rep are highly redundant. (The correlation between #utt and #rep is 0.91).

Given these predictions about the relative contribution of different factors to performance, it is then possible

[9] We assume an additive performance (utility) function because it appears that κ and the various cost factors c_i are utility independent and additive independent (Keeney and Raiffa, 1976). It is possible however that user satisfaction data collected in future experiments (or other data such as willingness to pay or use) would indicate otherwise. If so, continuing use of an additive function might require a transformation of the data, a reworking of the model shown in Figure 1, or the inclusion of interaction terms in the model (Cohen, 1995).

to return to the problem first introduced in Section 1: given potentially conflicting performance criteria such as robustness and efficiency, how can the performance of Agent A and Agent B be compared? Given values for α and w_i, performance can be calculated for both agents using the equation above. The mean performance of A is -.44 and the mean performance of B is .44, suggesting that Agent B may perform better than Agent A overall.

The evaluator must then however test these performance differences for statistical significance. In this case, a t test shows that differences are only significant at the $p < .07$ level, indicating a trend only. In this case, an evaluation over a larger subset of the user population would probably show significant differences.

2.5 Summary

We illustrated the PARADISE framework by using it to compare the performance of two hypothetical dialogue agents in a simplified train timetable task domain. We used PARADISE to derive a performance function for this task, by estimating the relative contribution of a set of potential predictors to user satisfaction. The PARADISE methodology consists of the following steps:

- definition of a task and a set of scenarios;
- specification of the AVM task representation;
- experiments with alternate dialogue agents for the task;
- calculation of user satisfaction using surveys;
- calculation of task success using κ;
- calculation of dialogue cost using efficiency and qualitative measures;
- estimation of a performance function using linear regression and values for user satisfaction, κ and dialogue costs;
- comparison with other agents/tasks to determine which factors that are most strongly weighted in the performance function generalize as important factors in other applications;
- refinement of the performance model.

Note that all of these steps are required to develop the performance function. However once the weights in the performance function have been solved for, user satisfaction ratings no longer need to be collected. Instead, predictions about user satisfaction can be made on the basis of the predictor variables, which is illustrated in the application of PARADISE to subdialogues in (Walker et al., 1997).

Given the current state of knowledge, many experiments would need to be done to develop a generalized performance function. Performance function estimation must be done iteratively over many different tasks and dialogue strategies to see which factors generalize. In this way, the field can make progress in identifying the relationships among various factors and can move towards more predictive models of spoken dialogue agent performance.

3 Discussion

In this paper, we reviewed the current state of the art in spoken dialogue system evaluation and argued that the PARADISE framework both integrates and enhances previous work. PARADISE provides a method for determining a performance function for a spoken dialogue system, and for calculating performance over subdialogues as well as whole dialogues. The factors that can contribute to the performance function include any of the cost metrics used in previous work. However, because the performance function is developed on the basis of testing the correlation of performance measures with an external validation criterion, user satisfaction, significant metrics are identified and redundant metrics are eliminated.

A key aspect of the framework is the decoupling of task goals from the system's dialogue behavior. This requires a representation of the task's information requirements in terms of an attribute-value matrix (AVM). The notion of a task-based success measure builds on previous work using transaction success, task completion, and quality of solution metrics. While we discussed the representation of an information-seeking dialogue here, AVM representations for negotiation and diagnostic dialogue tasks are also easily constructed (Walker et al., 1997). Finally, the use of κ means that the task success measure in PARADISE normalizes performance for task complexity, providing a basis for comparing systems performing different tasks.

4 Acknowledgments

Thanks to James Allen, Jennifer Chu-Carroll, Morena Danieli, Wieland Eckert, Giuseppe Di Fabbrizio, Don Hindle, Julia Hirschberg, Shri Narayanan, Jay Wilpon, and Steve Whittaker for helpful discussion on this work.

References

Abella, Alicia, Michael K Brown, and Bruce Buntschuh. 1996. Development principles for dialog-based interfaces. In *ECAI-96 Spoken Dialog Processing Workshop*, Budapest, Hungary.

Bates, Madeleine and Damaris Ayuso. 1993. A proposal for incremental dialogue evaluation. In *Proceedings of the DARPA Speech and NL Workshop*, pages 319–322.

Bernsen, Niels Ole, Hans Dybkjaer, and Laila Dybkjaer. 1996. Principles for the design of cooperative spoken human-machine dialogue. In *International Conference on Spoken Language Processing, ICSLP 96*, pages 729–732.

Carberry, S. 1989. Plan recognition and its use in understanding dialogue. In A. Kobsa and W. Wahlster, editors, *User Models in Dialogue Systems*. Springer Verlag, Berlin, pages 133–162.

Carletta, Jean C. 1996. Assessing the reliability of subjective codings. *Computational Linguistics*, 22(2):249–254.

Ciaremella, A. 1993. A prototype performance evaluation report. Technical Report Project Esprit 2218 SUNDIAL, WP8000-D3.

Cohen, Paul. R. 1995. *Empirical Methods for Artificial Intelligence*. MIT Press, Boston.

Danieli, M., W. Eckert, N. Fraser, N. Gilbert, M. Guyomard, P. Heisterkamp, M. Kharoune, J. Magadur, S. McGlashan, D. Sadek, J. Siroux, and N. Youd. 1992. Dialogue manager design evaluation. Technical Report Project Esprit 2218 SUNDIAL, WP6000-D3.

Danieli, Morena and Elisabetta Gerbino. 1995. Metrics for evaluating dialogue strategies in a spoken language system. In *Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 34–39.

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

Fraser, Norman M. 1995. Quality standards for spoken dialogue systems: a report on progress in EAGLES. In *ESCA Workshop on Spoken Dialogue Systems Vigso, Denmark*, pages 157–160.

Grice, H. P. 1967. Logic and conversation.

Grosz, Barbara J. and Candace L. Sidner. 1986. Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12:175–204.

Hirschberg, Julia and Christine Nakatani. 1996. A prosodic analysis of discourse segments in direction-giving monologues. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 286–293.

Hirschman, L., M. Bates, D. Dahl, W. Fisher, J. Garofolo, D. Pallett, K. Hunicke-Smith, P. Price, A. Rudnicky, and E. Tzoukermann. 1993. Multi-site data collection and evaluation in spoken language understanding. In *Proceedings of the Human Language Technology Workshop*, pages 19–24.

Hirschman, Lynette, Deborah A. Dahl, Donald P. McKay, Lewis M. Norton, and Marcia C. Linebarger. 1990. Beyond class A: A proposal for automatic evaluation of discourse. In *Proceedings of the Speech and Natural Language Workshop*, pages 109–113.

Hirschman, Lynette and Christine Pao. 1993. The cost of errors in a spoken language system. In *Proceedings of the Third European Conference on Speech Communication and Technology*, pages 1419–1422.

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

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

Krippendorf, Klaus. 1980. *Content Analysis: An Introduction to its Methodology*. Sage Publications, Beverly Hills, Ca.

Litman, Diane and James Allen. 1990. Recognizing and relating discourse intentions and task-oriented plans. In Philip Cohen, Jerry Morgan, and Martha Pollack, editors, *Intentions in Communication*. MIT Press.

Passonneau, Rebecca J. and Diane Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1).

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*, pages 28–33.

Price, Patti, Lynette Hirschman, Elizabeth Shriberg, and Elizabeth Wade. 1992. Subject-based evaluation measures for interactive spoken language systems. In *Proceedings of the DARPA Speech and NL Workshop*, pages 34–39.

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*, pages 49–54.

Siegel, Sidney and N. J. Castellan. 1988. *Nonparametric Statistics for the Behavioral Sciences*. McGraw Hill.

Simpson, A. and N. A. Fraser. 1993. Black box and glass box evaluation of the SUNDIAL system. In *Proceedings of the Third European Conference on Speech Communication and Technology*, pages 1423–1426.

Smith, Ronnie W. and Steven A. Gordon. 1997. Effects of variable initiative on linguistic behavior in human-computer spoken natural language dialog. *Computational Linguistics*, 23(1).

Smith, Ronnie W. and D. Richard Hipp. 1994. *Spoken Natural Language Dialog Systems: A Practical Approach*. Oxford University Press.

Sparck-Jones, Karen and Julia R. Galliers. 1996. *Evaluating Natural Language Processing Systems*. Springer.

Walker, Marilyn A. 1989. Evaluating discourse processing algorithms. In *Proc. 27th Annual Meeting of the Association of Computational Linguistics*, pages 251–261.

Walker, Marilyn A. 1996. The Effect of Resource Limits and Task Complexity on Collaborative Planning in Dialogue. *Artificial Intelligence Journal*, 85(1–2):181–243.

Walker, Marilyn A., Diane Litman, Candace Kamm, and Alicia Abella. 1997. Paradise: A general framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics, ACL/EACL 97*.