Empirical Methods for Evaluating Dialog Systems

Tim Paek
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
timpaek@microsoft.com

Abstract

We examine what purpose a dialog metric serves and then propose empirical methods for evaluating systems that meet that purpose. The methods include a protocol for conducting a wizard-of-oz experiment and a basic set of descriptive statistics for substantiating performance claims using the data collected from the experiment as an ideal benchmark or "gold standard" for making comparative judgments. The methods also provide a practical means of optimizing the system through component analysis and cost valuation.

1 Introduction

In evaluating the performance of dialog systems, designers face a number of complicated issues. On the one hand, dialog systems are ultimately created for the user, so usability factors such as satisfaction or likelihood of future use should be the final criteria. On the other hand, because usability factors are subjective, they can be erratic and highly dependent on features of the user interface (Kamm et al., 1999). So, designers have turned to "objective" metrics such as dialog success rate or completion time. Unfortunately, due to the interactive nature of dialog, these metrics do not always correspond to the most effective user experience (Lamel et al., 2000). Furthermore, several different metrics may contradict one another (Kamm et al., 1999), leaving designers with the tricky task of untangling the interactions or correlations between metrics.

Instead of focusing on developing new metrics that circumvents the problems above, we maintain that designers need to make better use of the ones that already exist. Toward that end, we first examine what purpose a dialog metric serves and then propose empirical methods for evaluating systems that meet that purpose. The methods include a protocol for conducting a wizard-of-oz experiment and a basic set of descriptive statistics for substantiating performance claims using the data collected from the experiment as an ideal benchmark or "gold standard" for making comparative judgments. The methods also provide a practical means of optimizing the system through component analysis and cost valuation.

2 Purpose

Performance can be measured in myriad ways. Indeed, for evaluating dialog systems, the one problem designers do *not* encounter is lack of choice. Dialog metrics come in a diverse assortment of styles. They can be subjective or objective, deriving from questionnaires or log files. They can vary in scale, from the utterance level to the overall dialog (Glass et al., 2000). They can treat the system as a "black box," describing only its external behavior (Eckert et al., 1998), or as a "glass box," detailing its internal processing. If one metric fails to suffice, dialog metrics can be combined. For example, the PARADISE framework allows designers to predict user satisfaction from a linear combination of objective metrics such as mean recognition score and task completion (Kamm et al., 1999; Litman & Pan, 1999; Walker et al., 1997).

Why so many metrics? The answer has to do with more than just the absence of agreed upon standards in the research community, notwithstanding significant efforts in that direction (Gibbon et al., 1997). Part of the reason deals with what purpose a dialog metric serves. Designers often have multiple and sometimes inconsistent needs. Four of the most typical needs are:

- Provide an accurate estimation of how well a system meets the goals of the domain task.
- Allow for comparative judgments of one system against another, and if possible, across different domain tasks.
- Identify factors or components in the system that can be improved.
- Discover tradeoffs or correlations between factors.

The above list of course is not intended to be exhaustive. The point of creating the list is to highlight the kinds of obstacles designers are likely to face in trying to satisfy just these typical needs. Consider the first need.

Providing an accurate estimation of how well a system meets the goals of the domain task depends on how well the designers have delineated all the possible goals of interaction. Unfortunately, users often have finer goals than those anticipated by designers, even for domain tasks that seem well defined, such as airline ticket reservation. For example, a user may be leisurely hunting for a vacation and not care about destination or time of travel, or the user may be frantically looking for an emergency ticket and not care about price. The "appropriate" dialog metric should reflect this kind of subtlety. While "time to completion" is more appropriate for emergency tickets, "concept efficiency rate" is more appropriate for the savvy vacationer. As psychologists have long recognized, when people engage in conversation, they make sure that they *mutually understand* the goals, roles, and behaviors that can be expected (Clark, 1996; Clark & Brennan, 1991; Clark & Schaefer, 1989; Paek & Horvitz, 1999, 2000). They evaluate the "performance" of the dialog based on their mutual understanding and expectations.

Not only do different users have different goals, they sometimes have multiple goals, or more often, their goals change dynamically in response to system behavior such as communication failures (Danieli & Gerbino, 1995; Paek & Horvitz, 1999). Because goals engender expectations that then influence evaluation at different points of time, usability ratings are notoriously hard to interpret, especially if the system is not equipped to infer and keep track of user goals (Horvitz & Paek, 1999; Paek & Horvitz, 2000).

The second typical need for a dialog metric – allowing for comparative judgments, introduces yet further obstacles. In addition to unanticipated, dynamically changing user goals, different systems employ different dialog strategies operating under different architectural constraints, rendering the search for dialog metrics that generalize across systems a lofty if not unattainable pursuit. While the PARADISE framework facilitates some comparison of dialog systems in different domain tasks, generalization is limited because different architectural constraints obviate certain factors in the statistical model (Kamm et al., 1997). For example, although the ability to "barge-in" turns out to be a significant predictor of usability, many systems do not support this. Task completion based on the kappa statistic appears to be a good candidate for a common measure, but only if every dialog system represented the domain task as an Attribute-Value Matrix (AVM). Unfortunately, that requirement excludes systems that use Bayesian networks or other non-symbolic representations. This has prompted some researchers to argue that a "common inventory of concepts" is necessary to have standard metrics for evaluation across systems and domain tasks (Kamm et al., 1997; Glass et al., 2000). As we discuss in the next section, the argument is actually backwards; we can use the metrics we already have to define a common inventory of concepts. Furthermore, with the proper set of descriptive statistics, we can exploit these metrics to address the third and fourth typical needs of designers, that of identifying contributing factors, along with their tradeoffs, and optimizing them.

This is not to say that comparative judgments are impossible; rather, it takes some amount of careful work to make them meaningful. When research papers describe evaluation studies of the performance of dialog systems, it is imperative that they provide a baseline comparison from which to benchmark their systems. Even when readers understand the scale of the metrics being reported, without a baseline, the numbers convey very little about the quality of experience users can expect of the system. For example, suppose a paper reports that a dialog system received an average usability score of 9.5/10, a high concept efficiency rate of 90%, and a low word error rate of 5%. The numbers sound terrific, but they could have resulted from low user expectations resulting from a simplistic interface. Practically speaking, to make sense of the numbers, readers either have to experience interacting with the system themselves, or have a baseline comparison for the domain task. This is true even if the paper reports a statistical model for predicting one or more of the dialog metrics from the others, which may reveal tradeoffs but not how well the system performs relative to the baseline.

To sum up, in considering the purpose a dialog metric serves, we examined four typical needs and discussed the kinds of obstacles designers are likely to face in finding a dialog metric that satisfies those needs. The obstacles themselves present distinct challenges: first, keeping track of user goals and performance expectations based on the goals, and second, establishing a baseline from which to benchmark systems and make comparative judgments. Assuming that designers equip their system to handle the first challenge, we now propose empirical methods that allow them to handle the second. These methods do not require new dialog metrics, but instead take advantage of existing ones through experimental design and a basic set of descriptive statistics. They also provide a practical means of optimizing the system.

3 Empirical methods

If designers want to make comparative judgments about the performance of a dialog system relative to another system so that readers unacquainted with either system can understand the reported metrics, they need a baseline. Fortunately, in evaluating dialog between humans and computers, the “gold standard” is oftentimes known; namely, human conversation. The most intuitive and effective way to substantiate performance claims is to compare a dialog system on a particular domain task with how human beings perform on the *same* task. Because human performance constitutes an ideal benchmark, readers can make sense of the reported metrics by assessing how close the system approaches the gold standard. Furthermore, with a benchmark, designers can optimize their system through component analysis and cost valuation.

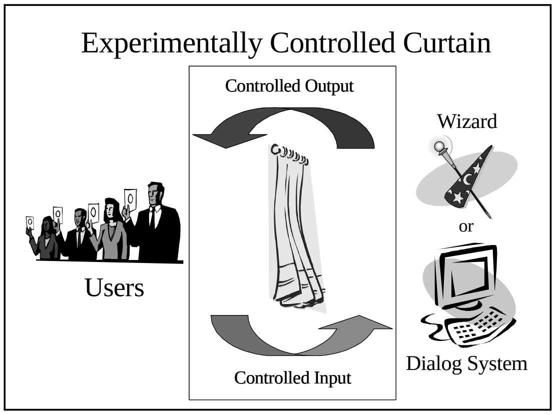

Figure 1. Wizard-of-Oz study for the purpose of establishing a baseline comparison.

In this section, we outline an experimental protocol for obtaining human performance data that can serve as a gold standard. We then highlight a basic set of descriptive statistics for substantiating performance claims, as well as for optimization.

3.1 Experimental protocol

Collecting human performance data for establishing a gold standard requires conducting a carefully controlled wizard-of-oz (WOZ) experiment. The general idea is that users communicate with a human “wizard” under the illusion that they are interacting with a computational system. For spoken dialog systems, maintaining the illusion usually involves utilizing a synthetic voice to output wizard responses, often through voice distortion or a text-to-speech (TTS) generator.

The typical use of a WOZ study is to record and analyze user input and wizard output. This allows designers to know what to expect and what they should try to support. User input is especially critical for speech recognition systems that rely on the collected data for acoustic training and language modeling. In iterative WOZ studies, previously collected data is used to adjust the system so that as the performance of the system improves, the studies employ less of the wizard and more of the system (Glass et al., 2000). In the process, design constraints in the interface may be revealed, in which case, further studies are conducted until acceptable tradeoffs are found (Bernsen et al., 1998).

In contrast to the typical use, a WOZ study for establishing a gold standard prohibits modifications to the interface or experimental "curtain." As shown in Figure 1, all input and output through the interface must be carefully controlled. If designers want to use previously collected performance data as a gold standard, they need to verify that all input and output have remained constant. The protocol for establishing a gold standard is straightforward:

- Select a dialog metric to serve as an objective function for evaluation and optimization.
- Vary the component or feature that best matches the desired performance claim for the dialog metric.
- Hold all other input and output through the interface constant so that the only unknown variable is who does the internal processing.
- Repeat using different wizards, making sure that each wizard follows strict guidelines for interacting with subjects.

To motivate the above protocol, consider how a WOZ study might be used to evaluate spoken dialog systems. As almost every designer has found, the "Achilles' heel" of spoken interaction is the fragility of the speech recognizer. System performance depends highly on the quality of the recognition. Suppose a designer is interested in bolstering the robustness of a dialog system by exploiting different types of repair strategies. Using task completion rate as an objective function, the designer varies the repair strategies utilized by the system. To make claims about the robustness of particular types of repair strategies, the designer must keep all other input and output constant. In particular, the protocol demands that *the wizard in the experiment must receive utterances through the same speech recognizer as the dialog system*. The performance of the wizard on the same quality of input as the dialog system constitutes the gold standard. The designer may also wish to keep the set of repair strategies constant while varying the use or disuse of the speech recognizer to estimate how much the recognizer alone degrades task completion rate.

A deep intuition underlies the experimental control of the speech recognizer. As researchers have observed, people with impaired hearing or non-native language skills still manage to communicate effectively despite noisy or uncertain input. Unfortunately, the same cannot be said of computers with analogous deficiencies. People overcome their deficiencies by collaboratively working out the mutual belief that their utterances have been understood sufficiently for current purposes, a process referred to as "grounding" (Clark, 1996). Repair strategies based on grounding indeed show promise for improving the robustness of spoken dialog systems (Paek & Horvitz, 1999; Paek & Horvitz, 2000).

3.1.1 Precautions

In following the above protocol, we point out a few precautions. First, WOZ studies for establishing a gold standard work best with dialog systems that are highly modular. The more modular the architecture of the dialog system, the easier it will be to test components by replacing a particular module of interest with the wizard. Without modularity, it will be harder to guarantee that all other inputs and outputs have remained constant because component boundaries are blurred. Ironically, after a certain point, a high degree of modularity may in fact preclude the experimental protocol; components may be so specialized and quickly accessed by a system that it may not be feasible to replace that component with a human.

A second precaution deals with the concept of a gold standard. What allows the performance of the wizard to be used as a gold standard is not the wizard, but rather the fact that the performance constitutes an upper bound. If an upper bound of performance has already been identified, then that is the gold standard. For example, graphical user interfaces (GUI) or touch-tone systems may represent a better gold standard for task completion rate if users finish their interactions with such systems ore often than with human operators. With spoken dialog systems, the question of when the use of speech interaction is truly compelling is often ignored. If a dialog designer runs the experimental protocol and observes that even human wizards cannot perform the domain task very well, that suggests that perhaps a gold standard may be found elsewhere.

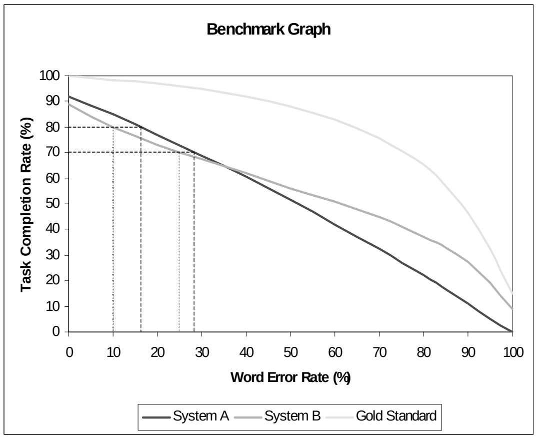

Figure 2. Comparison of two dialog systems with respect to the gold standard.

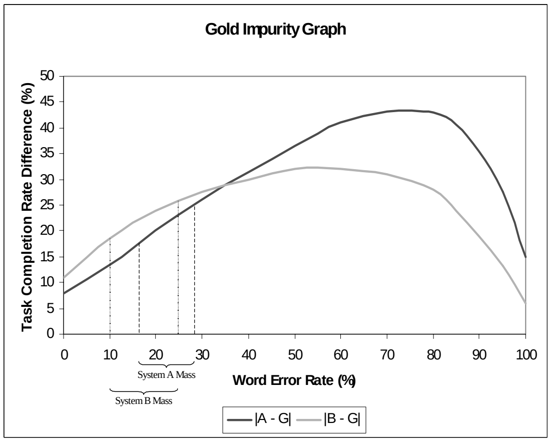

Figure 3. Distance in performance of the two systems from the gold standard.

3.2 Descriptive statistics

After collecting data using the experimental protocol, designers can make comparative judgments about the performance of their system relative to other systems with a basic set of descriptive statistics. The statistics build on the initial step of fitting a statistical model on the data fro both wizards and the dialog system. We discuss precautions later. Plotting the fitted curves on the same graph sheds light on how best to substantiate any performance claims. The graph displays the performance of the dialog system along a particular dimension of interest with the wizard data constituting a gold standard for comparison. Consider how this kind of "benchmark graph" could benefit the evaluation of spoken dialog systems.

Referring to previous example, suppose a designer is interested in evaluating the robustness of two dialog systems utilizing two sets of repair strategies. The designer varies which set is implemented, while holding constant the use of the speech recognizer. In general, as speech recognition errors increase, task completion rate, or dialog success rate, decreases. Not surprisingly, several researchers have found an approximately linear relationship in plotting task completion rate as a function of word error rate (Lamel et al., 2000; Rudnicky, 2000). Keeping this in mind, Figure 2 displays a benchmark graph for two dialog systems A and B, utilizing different repair strategies. The fitted curve for A is characteristically linear, while the curve for B is polynomial. Because wizards are presumably more capable of anticipating and recovering from speech recognition errors, their performance data comprise the gold standard. As such, the fitted curve for the gold standard in Figure 2 stays close to the upper right hand corner of the graph in a monotonically decreasing fashion; that is, task completion rate remains relatively high as word error rate increases and then gracefully degrades before the error rate reaches its highest level.

Looking at the benchmark graph, readers immediately get a handle on substantiating performance claims about robustness. For example, by noticing that task completion rate for the gold standard rapidly drops from around 65% at the 80% mark to about 15% by 100%, readers know that at 80% word error rate, even wizards, with human level intelligence, cannot recover from failures with better than 65% task completion rate. In short, the task is not trivial. This means that if A and B report low numbers for task completion rate beyond the 80% mark for word error rate, they may be still performing relatively well compared to the gold standard. Numbers themselves are deceptive, unless they are put side by side with a benchmark.

Of course, a designer might not have access to data all along the word error rate continuum as in Figure 2. If this presents a problem, it may be more appropriate to measure task completion rate as a function of concept error rate. The choice, as stated in the experimental protocol, depends on the performance claim a designer is interested in making. In spoken dialog, however, where speech recognition errors abound, another particularly useful benchmark graph is to plot word or concept error rate against user frustration. This experiment reveals any inherent bias users may have towards speaking with a computer in the first place.

In making comparative judgments, designers can also benefit from plotting the absolute difference in performance from the gold standard as a function of the same independent variable as the benchmark graph. Figure 3 displays the difference in task completion rate, or “gold impurity,” for systems A and B as a function of word error rate. The closer a system is to the gold standard, the smaller the “mass” of the gold impurity on the graph. Anomalies are easier to see, as they noticeably show up as bumps or peaks. If a dialog system reports low numbers but evinces little gold impurity, reader can be assured that the system is as good as it can possibly be.

Any crosses in performance can be revealing as well. For example, in Figure 3, although B performs worse at lower word error rates than A, after about the 35% mark, B stays closer to the gold standard. Hence, the designer in this case could not categorically prefer one system to the other. In fact, assuming that the only difference between A and B is the choice of repair strategies, the designer should prefer A to B if the average word error rate for the speech recognizer is below 35%, and B to A, if the average error rate is about 40%. Of course, other cost considerations come into play, as we describe later.

The final point to make about comparing dialog systems to a gold standard is that readers are able to substantiate performance claims across different domain tasks. They need only to look at how close each system approaches their respective gold standard in a benchmark graph, or how much mass each system puts out in a gold impurity graph. They can even do this without having the luxury of experiencing any of the compared systems.

3.2.1 Complexity

Without a gold standard, making comparative judgments of dialog systems across different domain tasks poses a problem for two reasons: task complexity and interaction complexity. Tutoring physics is a generally more complex domain task than retrieving email. On the other hand, task complexity alone does not explain what makes one dialog more complex than another; interaction complexity also plays a significant role. Tutoring physics can be less challenging than retrieving email if the system accepts few inputs, essentially constraining users to follow a predefined script. Any dialog system that engages in “mixed initiative” will be more complex than one that utilizes “system-initiated” prompts because users have more actions at their disposal at any point in time.

The way to evaluate complexity in a benchmark graph is to measure the distance of the gold standard to the absolute upper bound of performance. If wizards with human level intelligence cannot themselves perform reasonably close to the absolute upper bound, then either the task is very complex, or the interaction afforded by the dialog interface is too restrictive for wizards, or perhaps both. Because complexity is measured only in connection with the gold standard *ceteris paribus*, “benchmark complexity” can be computed as:

$$BC = n \cdot U - \sum_{x=0}^{n} g(x)$$

where U is the upper bound value of a performance metric, n is the upper bound value for an independent variable X, and $g(x)$ is the gold standard along that variable.

Designers can use benchmark complexity to compare systems across different domain tasks if they are not too concerned about discriminating between task complexity and interaction complexity. Otherwise, they can treat benchmark complexity as an objective function and vary the interaction complexity of the dialog interface to scrutinize the effect of task complexity on wizard performance, or vice versa. In short, they need to conduct another experimental study.

3.2.2 Precautions

Before substantiating performance claims with a benchmark graph, designers must exercise prudence in model fitting. One precaution is to beware of insufficient data. Without collecting enough data, designers cannot be certain that differences in the performance of a dialog system from the gold standard cannot be explained simply by the variance in the fitted models. To determine when there is enough data to generate reliable models, designers can conduct WOZ studies in an iterative fashion. First, collect some data and fit a statistical model. Second, plot the least squares distance, or $\sum_i (y_i - f(x_i))^2$, where $f(x)$ is the fitted model, against the iteration. Keep collecting more data until the plot seems to asymptotically converge. Designers may need to report R^2s for the curves in their benchmark graphs to inform readers of the reliability of their models.

Another precaution is to use different wizards, making sure that each wizard follows strict guidelines for interacting with subjects. The experimental protocol included this precaution because designers need to consider whether a consistent gold standard is even possible with a given dialog interface. Indeed, difference between wizards may uncover serious design flaws in the interface. Furthermore, using different wizards compels designers to collect more data for the gold standard.

As a final precaution, designers need to watch out for violations of model assumptions regarding residual errors. These are typically well covered in most statistics textbooks. For example, because task completion rate as a performance metric has an upper bound of 100%, it is unlikely that residual errors will be equally spread out along the word error rate continuum. In regression analysis, this is called "heteroscedasticity." Another common violation occurs with the non-normality of the residual errors. Designers would do well to take advantage of corrective measures for both.

3.2.3 Component analysis

A gold standard naturally lends itself to optimization. With a gold standard, designers can identify which components are contributing the most to a performance metric by examining the gold impurity graph of the system with and without a particular component. This kind of test is similar to how dissociations are discovered in neuroscience through "lesion" experiments. Carrying out stepwise comparisons of the components, designers can check for tradeoffs, and even use all or part of the gold impurity as an optimization metric. For example, suppose a designer endeavors to improve a dialog system from its current average task completion rate of 70% to 80%. In Figure 2, suppose B incorporates a component that A does not. Looking at the corresponding word error rates in the gold impurity graph for both systems, the mass under the curve for B is slightly greater than that for A. The designer can optimize the performance of the system by selecting components that minimize that mass, in which case, the component in B would be excluded. Because components often interact with each other in terms of their statistical effect on the performance metric, designers may wish to carry out a multi-dimensional analysis to weed out those components with weak main and interaction effects.

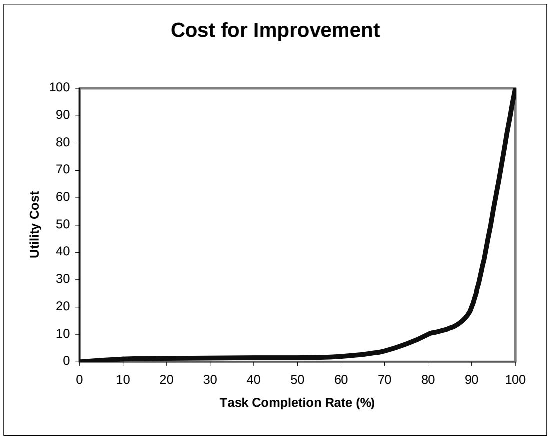

Figure 4. The cost a designer is willing to incur for improvements to task completion rate.

3.2.4 Cost valuation

Another optimization use of a gold standard is to minimize the amount of "gold" expended in developing a dialog system. Gold here includes more than just dollars, but time and effort as well. Designers can determine where to invest their research focus by calculating "average marginal cost." To do this, they must first elicit a cost function that conveys what they are willing to pay, in terms of utility, to achieve various levels of performance in a dialog metric (Bell et al., 1988). Figure 4 displays what cost a designer might be willing to incur for various rates of task completion. The average marginal cost can be computed by weighting gold impurity by the cost function. In other words, average marginal cost can be computed as:

$$AMC = \sum_{x=a}^{b} c(x) \cdot \left| f(x) - g(x) \right|$$

where $f(x)$ is the performance of the system on a particular dialog metric X, $g(x)$ is the gold standard on that metric, and $c(x)$ is elicited cost function.

Following the previous example, if the designer endeavors to improve a system that is currently operating at an average task completion rate of 70% to 80%, then the average marginal cost for that gain is simply the area under the cost function for that interval multiplied by the gold impurity for that interval. In deciding between systems or components, designers can exploit average marginal cost to drive down their expenditure.

4 Discussion

Instead of focusing on developing new dialog metrics that allow for comparative judgments across different systems and domain tasks, we proposed empirical methods that accomplish the same purpose while taking advantage of dialog metrics that already exist. In particular, we outlined an experimental protocol for conducting a WOZ study to collect human performance data that can serve as a gold standard. We then described how to substantiate performance claims using both a benchmark graph and a gold impurity graph. Finally, we explained how to optimize a dialog system using component analysis and value optimization.

Without a doubt, the greatest drawback to the empirical methods proposed is the tremendous cost of conducting WOZ studies, both in terms of time and money. In special circumstances, such as the Communicator Project, where participants all work within the same domain task, DARPA itself might finance WOZ studies for evaluation on behalf of the participants. Non-participants may resort to average marginal cost to optimize their own expenditure.

References

Bell, D. E., Raiffa, H., & Tversky, A. (Eds.). (1988). *Decision making: Descriptive, normative, and prescriptive interactions*. New York: Cambridge University Press.

Bersen, N. O., Dybkjaer, H. & Dybkjaer, L. (1998). *Designing interactive speech systems: From first ideas to user testing*. Springer-Verlag.

Clark, H.H. (1996). *Using language*. Cambridge University Press.

Clark, H.H. & Brennan, S.A. (1991). *Grounding in communication*. In Perspectives on Socially Shared Cognition, APA Books, 127-149.

Clark, H.H. & Schaefer, E.F. (1989). *Contributing to discourse*. Cognitive Science, 13, 259-294.

Danieli, M. & Gerbino, E. (1995). *Metrics for evaluating dialogue strategies in a spoken language system*. In Proc. AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation, 34-39.

Eckert, W., Levin, E. & Pieraccini, R. (1998). *Automatic evaluation of spoken dialogue systems*. In TWLT13: Formal semantics and pragmatics of dialogue, 99-110.

Gibbon, D., Moore, R. & Winski, R. (Eds.) (1998). *Handbook of standards and resources for spoken language systems*. Spoken Language System Assessment, 3, Walter de Gruyter, Berlin.

Glass, J., Polifroni, J., Seneff, S. & Zue, V. (2000). *Data collection and performance evaluation of spoken dialogue systems: The MIT experience*. In Proc. of ICSLP.

Horvitz, E. & Paek, T. (1999). *A computational architecture for conversation*. In Proc. of 7th User Modeling, Springer Wien, 201-210.

Kamm, C., Walker, M.A. & Litman, D. (1999). *Evaluating spoken language systems*. In Proc. of AVIOS.

Lamel, L., Rosset S. & Gauvain, J.L. (2000). *Considerations in the design and evaluation of spoken language dialog systems*. In Proc. of ICSLP.

Litman, D. & Pan, S. (1999). *Empirically evaluating an adaptable spoken dialogue system*. In Proc. of 7th User Modeling, Springer Wien, 55-64.

Paek, T. & Horvitz, E. (2000). *Conversation as action under uncertainty*. In Proc. of 16th UAI, Morgan Kaufmann, 455-464.

Paek, T. & Horvitz, E. (1999). *Uncertainty, utility, and misunderstanding: A decision-theoretic perspective on grounding in conversational systems*. In Proc. of AAAI Fall Symposium on Psychological Models of Communication, 85-92.

Rudnicky, A. (2000). *Understanding system performance in dialog systems*. MSR Invited Talk.

Walker, M.A., Litman, D., Kamm, C. & Abella, A. (1997). PARADISE: *A framework for evaluating spoken dialogue agents*. In Proceedings of the 35th ACL.