A Proposal for Incremental Dialogue Evaluation

Madeleine Bates and Damaris Ayuso

BBN Systems and Technologies
10 Moulton Street.
Cambridge, MA 02138

ABSTRACT

The SLS community has made progress recently toward evaluating SLS systems that deal with dialogue, but there is still considerable work that needs to be done in this area. Our goal is to develop incremental ways to evaluate dialogue processing, not just going from Class D1 (dialogue pairs) to Class D2 (dialogue triples), but measuring aspects of dialogue processing other than length. We present two suggestions; one for extending the common evaluation procedures for dialogues, and one for modifying the scoring metric.

INTRODUCTION

There is no single dialogue problem. By its nature, dialogue processing is composed of many different capabilities matched to many different aspects of the problem. It is reasonable to expect that dialogue evaluation methodologies should be multifaceted to reflect this richness of structure.

Ideally, each new addition to the set of evaluation methodologies should test a different aspect of dialogue processing, and should be harder than the methodologies that came before it. We present two suggestions: one which extends the common evaluation procedure in order to test one new aspect of dialogues, and one which modifies the scoring metric.

An Analogy with Chess

We as a community have been thinking about dialogue evaluation in terms of whether the systems we are building give the "right" answer (the one the wizard gave, or the one agreed upon by the Principles of Interpretation) at every step. We have been trying to come up with a methodology to measure whether our systems can reproduce the wizard's answers at each step of a lengthy dialogue. But is this a reasonable approach?

Participating in a dialogue, whether between two humans or between a human and a machine, bears a striking resemblance to playing a complex game such as chess.

Similarities:

1. Each involves precise turn-taking.
2. There is an extremely large tree of possible "next moves" (the tree for human dialogue, even in a limited domain, is much larger than that for chess).
3. Multiple paths through the tree can lead to the same results.

Differences:

1. Conversation is cooperative, but a game is competitive.
2. In chess, the goal is clear (checkmate), but in a conversational dialogue, the goal is less clear.
3. In a chess game, any state can be completely and concisely represented by a single board position; in a dialogue it is not known what comprises a state, nor how to represent it.

Like the game tree for chess, the human/computer dialogue tree is enormous, as indicated in figure 1. There are usually hundreds or thousands of alternatives the human may produce. The number of responses the system can make is much smaller; some responses may be clearly wrong, but seldom is there a single "right" or "best" response (just as there is seldom a single such move in chess). Even when striving for the same goal, two different people are very likely to choose very different paths.

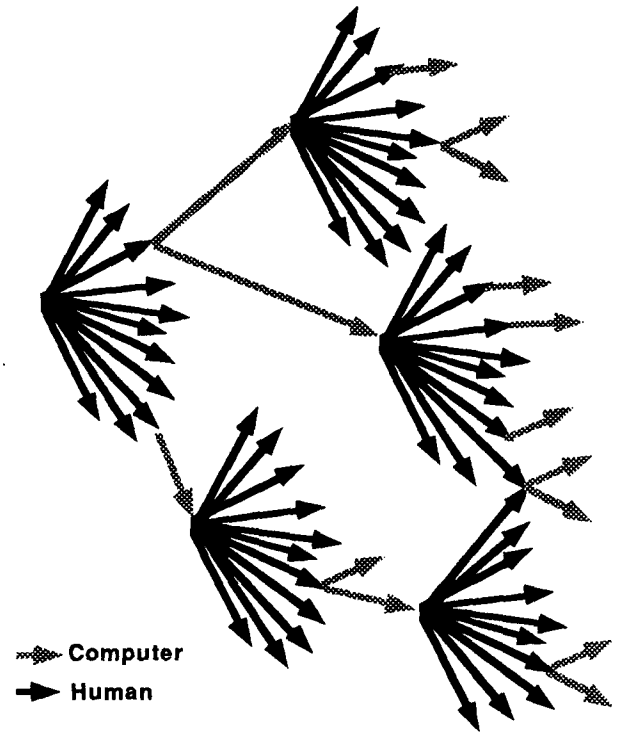

Figure 1: A Human/Computer Dialogue Tree

As a community, we have been thinking about dialogue evaluation in terms of whether the system gives the "right" answer at every step (the one the wizard gave at the same point in the same dialogue). The major problem with this type of thinking is that it encourages us to characterize a move that does not mimic the expert's (or an answer that does not exactly match the wizard's) as wrong, when it may not be wrong at all, but just different.

WHY EVALUATING DIALOGUE SYSTEMS BY SEEING WHETHER THEY REPRODUCE WIZARD SCENARIOS IS A BAD IDEA

We have been trying to think about dialogue evaluation in terms of measuring whether our systems can reproduce virtually the same answers that the wizard produced for an entire dialogue sequence. This is like asking one chess expert to exactly reproduce every move that some other expert made in a past game!

There are several reasons why this cannot work in the human-computer environment either. We will briefly discuss three of these: ambiguous queries, failure to answer by the system, and wrong interpretations by the system.

Ambiguous Queries

Ambiguous queries abound in the training sentences, and will certainly appear in test sets as well. As soon as a system can give a valid answer that is different from the wizard's, there is the potential for a significant divergence in the paths taken by the wizard's session and the system's execution of the dialogue.

This means that the query following the ambiguous query in the test set (i.e., from the wizard's session) may not make sense as a follow-up query when the system processes it! Consider the following examples:

Q1: Show me flights from San Francisco to Atlanta.
Q2: Show me flights that arrive after noon.
Q3: Show me flights that arrive before 5pm.

A3' (List of flights between noon and 5pm)

A3" (List of flights before 5pm, including morning arrivals)

Q3: Show me the fare of AA123.

The answer A3" is a superset of the answer A3'. Therefore if the wizard answers as in A3", but a system being evaluated produces A3', and the flight referred to in Q3 is in A3" but not A3', then the answer produced by the system is likely to be officially "wrong", although it is perfectly correct, given the actual context available to the system at the time Q4 was processed.

Of course, some ambiguities in the class A sentences in this domain may in practice not affect the course of the dialogue this way. However, in extended dialogues with context-dependencies, substantial ambiguities can quickly develop.

No Answer

Another source of dialogue path divergence occurs when the system is unable to answer a query which the wizard answered. For example:

Q1: Show me flights from San Francisco to Atlanta.
Q2: Which is the cheapest one if I want to travel Monday.
Q3: What is the fare on that flight?
Q4: Show me its restrictions.

If the system were unable to understand Q2, producing no answer, then Q3 and Q4 would not be understandable at all.

But what would happen in a real human-computer dialogue (instead of the highly artificial task of trying to copy a pre-specified dialogue)? If the system clearly indicated that it didn't understand Q2 at all, a real live user with even moderate intelligence would never ask Q3 at all, because s/he would realize that the system did not have the proper context to answer it. Instead, s/he might continue the dialogue quite differently, for example:

Q3: Give me the restrictions and fare of the cheapest Monday flight.

Notice that by following this alternate path, the user may actually be able to get the data s/he ultimately wants sooner than the wizard scenario provided it (in 3 queries instead of 4, even though one of those 3 was not understood). How can one say that such a system is less good than one that mimics the wizard perfectly in this case?

Wrong Interpretation

As in the previous two cases, an incorrect interpretation by the system causes a divergence in the dialogue tree. In addition, as in the No Answer case, in practice a wrong interpretation is likely to result in a glaring error that elicits follow-up clarification queries from the user. Even if the user does not follow up in exactly that way, s/he will take into account that the system did something unexpected, and will use that knowledge in the formulation of subsequent queries. Just as in the No Answer case, the dialogue branch followed after a wrong interpretation could easily be fundamentally different than the branch which the wizard would follow, but may lead to the same point (all the information the user wanted).

What does it mean?

Our goal should not be to produce systems that behave exactly like the wizard, but rather systems that respond reasonably to every input and that allow the user to reach his or her goal.

Perhaps if we give up the notion of "the right answer" in favor of "a reasonable answer" then we can develop more effective and meaningful evaluation methodologies. The following two proposals provide some concrete suggestions as to how to go about this process. The first deals with the aspect of breadth in the dialogue tree, the second deals with reasonable responses to partially understood input.

PROPOSAL: DIALOGUE BREADTH TEST

One of the central problems of dialogue evaluation is that there is no single path of questions and answers to which a system must conform,but rather a plethora of possible questions at each point in the dialogue. A very important capability of a good system is to be able to handle many different queries at each point in a dialogue.

We suggest here a methodology that deals with one important aspect of dialogue; specifically, it attempts to compare systems on their ability to handle diverse utterances in context. This method builds on our existing methodology without imposing unrealistic constraints on the system design. It also meets the requirement for objective evaluation that it must be possible to agree on what constitutes a reference answer.

Consider a dialogue that begins with two queries. At this point, if you ask ten different people what question they would ask next, you will likely get ten different queries, as illustrated in figure 2.

Context:

Q1: I want to go from Boston to Dallas on Saturday.
A1: (A list of flights and associated information)
Q2: Which of those flights are nonstops?
A2: (A smaller list of flights and other information)

Some possible next queries:

Q3a: How much do they cost?
Q3b: Which one has the lowest fare, no restrictions?
Q3c: What's the cost of a coach ticket on AA123?
Q3d: Which flight is cheapest?
Q3e: Now show me flights Dallas to Boston.
Q3f: What is the cost of taxis in Dallas?
Q3g: Is flight AA123 a DC-10?
Q3h: Just the ones that leave before noon, please.

Figure 2: An Example of Dialogue Breadth

It is reasonable to ask how many of these natural, possible dialogue continuation utterances a particular NL system can understand, and it is also reasonable to compare one system with another based on which one can handle more of the possible continuation utterances.

In this type of evaluation, the object is to control the initial part of a dialogue to that which a number of different systems can handle, and then to see how many possible alternative utterances (from a given set) each system could handle at that point in the dialogue. Of course, the continuation utterances should be as meaningful and natural as possible and should not necessarily be context dependent, as in Q3e and Q3f.

Dialogue Breadth Test Methodology

We present here an example of how an evaluation to assess a system's capabilities in handling dialogue breadth would take place. We call this test the Dialogue Breadth (DB) test. The numbers and other details are for illustrative purposes only.

Each site would be given a set of 15 dialogue starters (initially, let's assume that is 15 Q1 utterances). With each Q1, there would be a set of 10 alternative Q2 utterances; this would form a test set of 150 Q1-Q2 dialogue pairs. Sites would run the complete dialogues through their systems, and would return 100 test items and answers for scoring. (The reduction from 150 would enable sites to remove many cases in which Q1 was not processed correctly, thus focussing the test on the issue of dialogue analysis, not Class A processing.) These 100 answers would be automatically compared to reference answers and scores computed in the usual way: as the number (and percentage) of the continuation utterances that were answered correctly, incorrectly, or not answered at all.

How would the test sets of "next utterances" for the test set be obtained? It could easily be done just as described above -- by showing at least 10 people the original problem scenario that the original wizard subject was trying to solve, showing them the initial dialogue context, and then asking them to add a single utterance.

The first time the evaluation is tried, it might make sense to use a single Class A utterance as the context, as described above, and consider the breadth that is possible at just the second step of a dialogue. Later evaluations could be carried out on slightly longer dialogue fragments, where each context is a pair of Q1 and Q2 utterances, followed by a set of 10 alternate Q3 utterances.

Advantages of the Dialogue Breadth Test

There are a number of advantages to be gained by adding the dialogue breadth test to the SLS community's growing set of evaluation tools.

1. This methodology builds on the Class D methodology which the community has developed thus far.

2. It examines an extremely important aspect of dialogue systems: the ability to handle a variety of utterances at a point in mid-dialogue.

3. As long as short dialogue contexts are used, it does not depend on each site building systems that try to duplicate the output of the wizard, since intermediate answers to the initial dialogue utterances are not scored.

4. It requires no more training data than our current dialogue evaluation mechanism. Although it would be useful for sites to have a small amount of dialogue breadth training data, most system development can proceed using the data that has already been collected, or more data of that type.

5. It requires no changes to the classification scheme, or to other information associated with the training data, such as the reference answers.

SUGGESTION: MODIFY METRICS TO ENCOURAGE PARTIAL UNDERSTANDING

In human-human dialogue (let us call the participants the questioner and the information agent), it is often the case that one party only partially understands the other, and realizes that this is the case. There are several things that the information agent can do at this point:

1. ask for clarification

2. provide an answer based on partial understanding

3. provide an answer based on partial understanding, while indicating that it may not be precisely what was desired

4. decline to answer, and make the other party ask again.

Clearly, 2 and 4 are the least preferred responses, since they may mislead or frustrate the questioner, respectively. But both 1 and 3 are often reasonable responses.

For example, if an information agent hears,

"What are the flights from BOS to DFW that ..... lunch"

(where some language in the middle was not heard or understood for some reason), it is reasonable to respond with either a request for clarification, such as,

"Do you want BOS to DFW flights that serve lunch?"

or with a qualified answer, such as,

"I didn't entirely understand you, but I think you were asking for BOS to DFW flights with lunch, so here they are:
TWA 112 ...".

Either response is acceptable, even to a questioner who asked for flights "that do not serve lunch", or flights "that serve breakfast or lunch" because the system made clear that its understanding was uncertain.

The idea of allowing our SLS systems to respond, in effect, with an answer qualified by "I'm not sure I caught everything you said, but here's my best guess" is a powerful one that is clearly oriented toward making systems useful in application.

A Suggestion for Change

The need for permitting systems some leeway in responding to partially understood input is clear, but the mechanism for doing so is less clear, and would require some thought by all of those involved in developing the SLS evaluation methodology.

For example, a new class of system response could be allowed, called "Qualified Answer", and that two new categories called Qualified Answer Reasonable, Qualified Answer Not Reasonable be added to the current set of Right, Wrong, and No Answer.

Judging a qualified answer as reasonable or unreasonable would almost certainly have to be done by a human judge or judges, since it is unlikely to be possible to anticipate all the possible reasonable answers to a query. It would also be necessary to develop an explicit scoring metric for the new categories which would not penalize qualified answers too harshly.

The evaluation committee and the SLS steering committee should consider these suggestions for possible inclusion in future common SLS evaluations.

ACKNOWLEDGEMENTS

The work reported here was supported by the Advanced Research Projects Agency and was monitored by the Office of Naval Research under Contract No. N00014-89-C-0008. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research projects Agency or the United States Government.

REFERENCES

1. Bates, M., Boisen, S., and Makhoul, J. "Developing and Evaluation Methodology for Spoken Language Systems", *Proceedings of the Speech and Natural Language Workshop* (June, 1990), Morgan Kaufmann Publishers, Inc., 1990.

2. Hemphill, C.T., Godfrey, J.J., and Doddington, G.R. "The ATIS Spoken Language Systems Pilot Corpus", *Proceedings of the Speech and Natural Language Workshop* (June, 1990), Morgan Kaufmann Publishers, Inc., 1990.

3. Hirschman, L., Dahl, et al "Beyond Class A: A Proposal for Automatic Evaluation of Discourse", *Proceedings of the Speech and Natural Language Workshop* (June, 1990), Morgan Kaufmann Publishers, Inc., 1990.

4. Pallett, D.S., Fisher, W.M., Fiscus, J.G., and Garofolo, J.S. "DARPA ATIS Test Results", *Proceedings of the Speech and Natural Language Workshop* (June, 1990), Morgan Kaufmann Publishers, Inc., 1990.

5. Price, P. "Evaluation of Spoken Language Systems: the ATIS Domain", *Proceedings of the Speech and Natural Language Workshop* (June, 1990), Morgan Kaufmann Publishers, Inc., 1990.