AN OPEN CONCEPT METRIC FOR ASSESSING DIALOG SYSTEM COMPLEXITY

Thomas M. DuBois and Alexander I. Rudnicky

School of Computer Science,
Carnegie Mellon University,
Pittsburgh, PA, USA

ABSTRACT

Techniques for assessing dialog system performance commonly focus on characteristics of the interaction, using metrics such as completion, satisfaction or time on task. However, such metrics are not always capable of differentiating systems that operate on fundamentally different principles, particularly when tested on tasks that focus on common-denominator capabilities. We introduce a new metric, the *open concept count*, and show how it can be used to capture useful system properties of a dialog system.

1. INTRODUCTION

Dialog systems can either be either highly directed, only listening for the user to respond to what was just asked, or they can support a mixed-initiative dialog, prompting the user for input, but allowing the user to either respond to the prompt or to change to a different topic. When evaluating on tasks that only occasionally requires the system to accommodate a change in topic [3], it can be difficult to note differences between systems that are based on such different underlying architectures. Yet the ability to respond correctly to changes in topic differentiates systems that gracefully accommodate shifting user goals (a characteristic of complex problem-solving domains) from those that do not. We propose to quantify this distinction by computing the number of possible inputs, or *open concepts*, that a system can respond to on any one turn. We further propose that this be used as a measure of the inherent complexity of a dialog system.

The core of the approach is to measure the number of different inputs that can be understood by the system at a specific point in a dialog. We can capture this property in the following equation:

$$C_x = \frac{1}{t}\sum_{i}^{t} C(c_i)$$

where the complexity of a system is characterized by the number of concepts available per turn, *c*, averaged over the turns, *t*, in a dialog with the system. A mixed-initiative system would exhibit a higher open-concept count than a directed-dialog system.

This metric can be used to provide an overall characterization of a particular dialog system. It can also be used to produce a detailed analysis of system performance. For example, the open concept count also allows us to quantitatively measure a system's ability to *acquire* information, as well as the ability of a system to *accommodate* to what the user may say in a particular context.

2. OPEN CONCEPTS

For purposes of the current work, we operationally define concepts as corresponding to unique slots in the system's semantic grammar (which in turn correspond to entities in the task domain) that the dialog system is capable of understanding (that is, parse and possibly act upon). Certain concepts in the domain can be further differentiated into a range of unique values, such as (in the travel domain) city names or dates. We do not further consider the number of levels within a base concept for purposes of the current discussion, though taking this aspect into account is likely an important aspect of fully characterizing the openness of a given system but one that more properly features in a discussion of domain coverage rather than system complexity per se.

An open concept is any concept that can be understood and acted upon at a given point in a dialog. Most simply this could mean that the system contains an action rule in which the input concept is featured as part of the conditional. In a directed dialog system, we would expect the number of open concepts at any point in time to be small and constant over the course of the dialog. For example, the system might be responsive to the concept(s) queried for in the prompt, plus additional global items such as HELP, REPEAT or MAIN_MENU. A mixed-initiative system on the other hand might be open to a comparatively large number of concepts at any point in time, accepting arbitrary (in domain) inputs from the user. In practice, a sophisticated dialog system will operate in

both modes, dynamically scoping the number of acceptable concepts based on context and history.

3. INFORMATION ACQUISITION

Openness per se is insufficient for characterizing system behavior; we also need to consider how it interacts with the process of acquiring task information from the user. Dialog systems designed to perform complex tasks achieve their primary goal by decomposing it into sub-goals. For example, a travel planning system, in order to arrange a flight, must accomplish the sub-goals of finding out when and where the flight should leave as well as where it is going. We measure system progress by completion of sub-goals per turn. Any particular system's definition of sub-goals is somewhat arbitrary. However, provided there is some consistency throughout the system, this is not really an issue.

4. ACCOMODATION

Degree of accommodation, or the ability of the system to accommodate the user stepping outside of its questioning, can be indicated in two ways. First, any instance in which the user says something that the system could parse, but gets discarded because the system is not open to the input, is a direct example of the system not being accommodating enough. Unfortunately this is difficult to track because a system may not differentiate this case from, e.g., out-of-domain inputs. Another possibility is to measure how often the system parses, and makes use of, something other than what it was prompting for, or when the system parses the user's statement into multiple concepts, and is able to make use of these. These are clearly examples of the user wanting to go outside of the framework suggested by the system, and the system being able to accommodate the departure.

5. SYSTEM STATE

Regardless of the overall openness of the system, we need to have a meaningful definition of what state the system is in on any given turn and to associate an open concept count with that state. We have come up with an intuitive and highly generalizable way to do this. The set of states is based on the type of information the system is trying to convey to, or collect from the user. For example, *"on what day will you be leaving Pittsburgh?"* is the same state as *"a flight from Washington dc, on what date will you be leaving?"* because they both are trying to collect departure dates. However questions such as *"are you a registered user?"* and *"do you want to be emailed this itinerary"* are two separate states even though they both call for yes or no answers. Since this definition is based on what the system is expecting in a given turn, it is particularly good for measuring both how well the system can gather information it wants, and how well it can accommodate information other than that which it asked for. The Communicator system has 43 such states, although just 10 of these states make up 75% of the dialog.

6. CASE STUDY: CMU COMMUNICATOR

We explored the open concept metric by using the CMU Communicator system [1], a telephone-based travel planning system using live schedule data. The Communicator accepts airline and hotel requests and interactively constructs itineraries for the user. The CMU Communicator manages dialog using the Agenda architecture [2]. The system was instrumented to log all open concepts available on a given turn by traversing the agenda and noting all concepts that the system could respond to. The analyses reported in this paper were carried out on a corpus collected January-May 2001 and include 476 calls, with a total of 8474 turns (calls that did not proceed to the acceptance of the first leg of the flight itinerary and those where there was some doubt as to the correct alignment of the log files were excluded from the corpus).

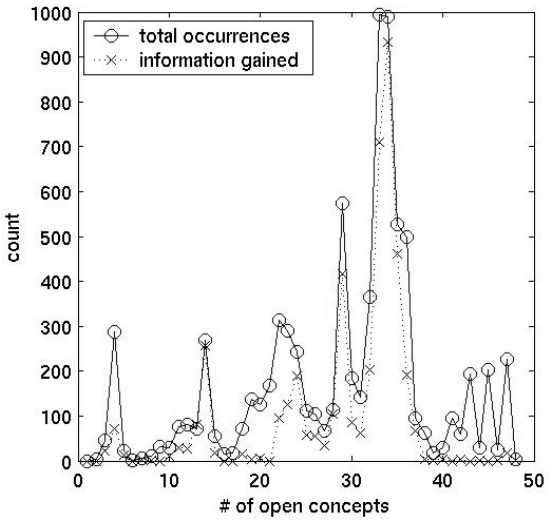

Figure 1. *The solid line interpolating the circles shows on how many turns (y-axis) a particular number of concepts is open (x-axis) throughout the corpus. The dashed line interpolating the x's shows how much total information was acquired by the system (y-axis) for a given number of open concepts.*

Some examples of concepts the system listens for are: *user name*, *departure airport*, *goodbye*, *help*, *summary*,

and *hotel name*. What this means is that if the concept *departure airport* is open on a particular turn, and the user says where he wants to fly from, the system will parse it correctly. If however the concept is not open, the system will not be able to correctly interpret what was said. Which concepts are open on every turn is logged by the system. Also, each sub-goal accumulated on a turn is logged as part of an itinerary. System state at any turn is parsed out from a log of the dialog manager. This system is designed to be mostly open, becoming directed only when repeated prompts for some information fail. Therefore, most of the time, it exhibits a large number of open concepts (the peak of the largest mode is at 33).

6.1. Modes of operation

Figure 1 shows how often a given number of open concepts occur, overlaid with a plot of how much information is gained with a given number of open concepts. Figure 1 shows a profile of the system in terms of the frequency of states of a given degree of openness as well as the number of sub-goals acquired in that state. The system tends to operate in only a few major modes, each one corresponding to a question, or group of questions that the system can ask. The bulk of the system's time, and also information gain, occurs with between 30 and 40 concepts open. This is where most flight related questions are asked. Since flight related questions are the first thing the system asks about, it makes sense that a high number of concepts are open at the time, since it is listening for everything it needs to construct an itinerary.

The second major group of open concepts occurs between 20 and 27, and is mostly related to discussion of hotel and rental car arrangement, which usually happens closer to the end of a call, after the system has most of the information it needs.

6.2. Correspondence between prompt and input

Given that we have a way to describe the calls by openness, we need a way to determine user accommodation. The first way we do this is to look at how often the user wants to provide more than one piece of information. Out of 3581 turns where some information gain was present, during 23.5% of them the user gave the system more than one piece of information. This means that if the system were limited to listening for one thing at a time, a significant portion of the time it would force users to move more slowly than they want. While this might not be a major issue for one-time users, a more practiced user familiar with the task might find this confining.

The second way to measure accommodation is to look at how often the system asks for something other than a particular sub-goal, and yet the user's response accumulates that sub-goal. An example would be if the system says *"Do you really want to start over?"* which is clearly not goal related, and the user responds with *"No, I want to fly from Miami,"* which is goal related. The simplest way to measure this is to look at all the turns where the system says something non-goal related, and count how much information is accumulated. In our analysis, Communicator says something of this type 3272 times in our data set, and gains 685 pieces of information. This is approximately 17% of the total information the system collects. While this is not a high rate of acquisition, there is still quite a difference from the (0) rate at which a fully directed system would operate.

6.3. Example modes

We can look at a specific question that the system asks. One of the first things asked, is *"What is your full name?"* Usually at this point the system has about fourteen concepts open, which amounts to not much more than the standard help concepts (see Figure 2).

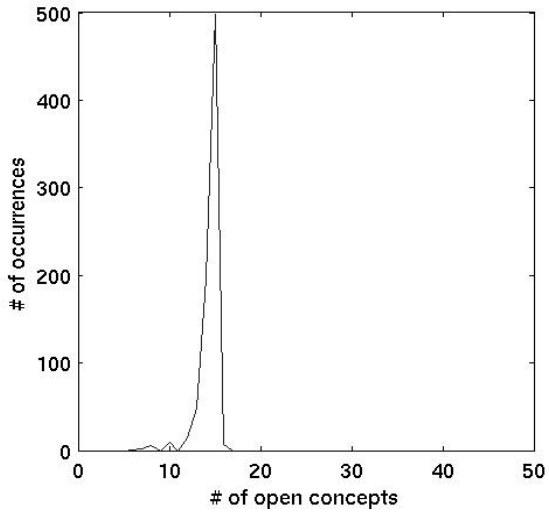

Figure 2. *This graph shows only at instances from the corpus where the system asked a variant of "what is your full name?" Within this constraint, how many turns have a given number of open concepts is plotted as a function of open concepts.*

For reasons including the specific focus of this question, and its ease, it is answered a higher percentage of the time than any other question, at 92%. However, very rarely does the system gain multiple pieces of information after this question. Less than 9% of the time the question is answered is any additional information provided. This is especially low considering that this question occurs towards the beginning of a call where the user has a lot of information still to convey to the system. Also only 6% of the time this question is asked does the

system have to repeat it. This error rate puts it at about the system's ideal.

To show how this differs from a state with many open concepts, we will contrast this state with the one in which the system asks, *"To where are you traveling?"* (see Figure 3). This state's peak is around 34 concepts open. As with the last one, this question is prompting for a very specific answer, and yet it is more accommodating. Of the 488 turns out of 597 in which this question gains information, 22% of the time; it gains multiple pieces of info instead of just one. As a tradeoff this question has a higher repetition rate than the last one, but 15% is still close to the system's best, and some of the difference can be attributed to it being a slightly more complicated question than *"What is your name."*

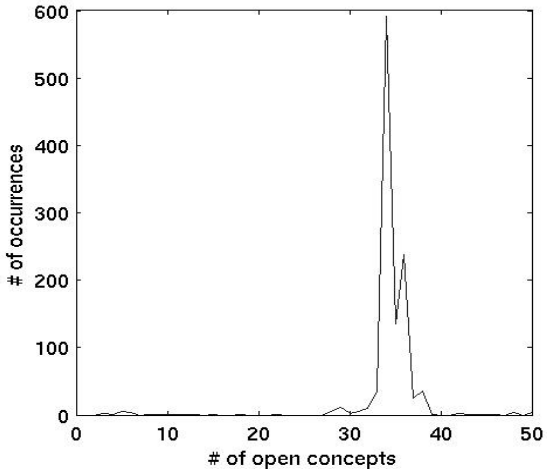

Figure 3. *Same as figure 2, only the question "to where are you traveling?" is considered instead. Number of occurrences is plotted against number of open concepts.*

Question name	Turns	% answered	% over answered	% repetition
Arrive City	597	81.7	21.9	15.6
Username	344	92.1	8.5	6.4

Figure 4. Arrive City peaks at 34 open concepts and is an example of a more mixed-initiative state, while Username peaks at 14 and is an example of a directed state.

For the CMU Communicator, states with more open concepts appear to produce greater user accommodation while those with fewer open concepts appear to have a higher use compliance rate, though this appears to be conditioned on the specific context (see Figure 4).

7. CONCLUSION

Directed and mixed initiative dialog systems operate in fundamentally different ways. To examine the effect of this difference we propose a quantitative measure, *open concept count*. The open concept count provides a simple and direct characterization of the inherent complexity of the system's behavior. Complexity as defined here directly affects the ability of a system of a system to accommodate mixed-initiative dialog, which in turn is required for complex problem solving behavior. The open concept count can be easily computed from logs and provides an insight into a system's behavior. It would be useful to see this type of analysis applied to other dialog systems and determine if the results obtained here are specific to our system, or indicative of a more general property dialog systems.

8. ACKNOWLEDGEMENTS

We would like to thank Wei Xu for the initial implementation of the logging mechanism in the CMU Communicator system. This research was sponsored in part by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

9. REFERENCES

[1] Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu W., Oh, A. Creating natural dialogs in the Carnegie Mellon Communicator system. Proceedings of Eurospeech, 1999, 4, 1531-1534

[2] Rudnicky, A. and Xu W. An agenda-based dialog management architecture for spoken language systems. IEEE Automatic Speech Recognition and Understanding Workshop, 1999, p I-337.

[3] M. Walker et al. DARPA Communicator Dialog Travel Planning Systems: The June 2000 Data Collection. Eurospeech 2001, *these proceedings*.