

Elements of a Computational Model of Cooperative Response Generation*

Brant A. Cheikes and Bonnie L. Webber
University of Pennsylvania
Department of Computer and Information Science
Philadelphia, PA 19104-6389
brant@linc.cis.upenn.edu
bonnie@central.cis.upenn.edu

March 9, 1989

Abstract

If natural language question-answering (NLQA) systems are to be truly effective and useful, they must respond to queries *cooperatively*, recognizing and accommodating in their replies a questioner's goals, plans, and needs. This paper concerns the design of *cooperative response generation* (CRG) systems, NLQA systems that are able to produce integrated cooperative responses. We propose two characteristics of a computational model of cooperative response generation. First, we argue that CRG systems should be able to explicitly reason about and choose among the different response options available to them in a given situation. Second, we suggest that some choices of response content motivate others—that through a process called *reflection*, respondents detect the need to explain, justify, clarify or otherwise augment information they have already decided to convey.

1 Introduction

Our success in day-to-day affairs depends to a great extent on the cooperation of those with whom we interact. Studies of man-machine interaction show that we expect the same from the (complex) systems we deal with [14]. Here we consider the case of natural language question-answering (NLQA) systems. To be cooperative, any system must recognize and accommodate the goals, plans, and needs of its users. For NLQA systems, this means that they must take the initiative when responding, rather than answering queries passively. They cannot simply translate input queries into transactions on database or expert systems—they have to apply many more complex reasoning mechanisms to the task of deciding how to respond. It has been suggested that cooperative NLQA systems must be able to provide *extended responses* [15, 16], combining such elements as:

- a direct answer;
- information or action that is pertinent to the direct answer;
- information or action pertinent to one or more of the questioner's stated or inferred goals.

Cooperative behavior in natural language has been studied by many researchers. At the University of Pennsylvania alone, several projects have focused on different aspects of cooperative response production,

*This research was partially supported by DARPA grant N0014-85-K0018.

including Kaplan's work on responding when presuppositions fail [7], Mays' work both on responding when queries fail intensionally [8] and on determining competent monitor offers [9], McKeown's TEXT system for explaining concepts known to a database system [11], McCoy's system for correcting object-related misconceptions [10], Hirschberg's work on scalar implicatures and their use in avoiding the production of misleading responses [5], and Pollack's plan inference model for recognizing and responding to discrepancies between the system's and the user's beliefs about domain plans and actions [12]. Other explorations of cooperative communication include [1], [2], [4], [6], [13], [15], and [17]. For more complete references, the reader is referred to Cheikes [3].

The results of these studies have been highly informative. Many different kinds of cooperative behavior have been identified, and computational models of them proposed. What is of interest to us here is the fact that all efforts to date in this area share the same implicit assumption—that cooperative response generation can be decomposed into separate reasoning processes. But this “decomposability assumption” in turn raises the *integration problem*—the problem of getting those elements to work together in the production of a single response. This so far has largely been ignored.

Solving the integration problem means devising an architecture for *cooperative response generation* (CRG) systems—NLQA systems that can combine in their responses instances of different kinds of cooperative behavior appropriate to the situation. Now that the study of cooperativeness in natural language is beyond its infancy (although still far from mature), it is an appropriate time to confront the integration problem and study the design of CRG systems. This paper describes the beginnings of such a computational model of CRG.

2 Toward a Model of Cooperative Response Generation

What characteristics of the CRG process might be used to motivate the design of a CRG system? Analysis of various transcripts of natural language question-answer dialogues leads us to propose two: (1) CRG systems should be able to explicitly reason about and choose among the different response options available to them in a given situation, and (2) CRG systems should be able to *reflect* on their selections of response content before producing any output. Some choices of response content motivate others, and the process of reflection allows respondents to detect a need to explain, justify, or clarify information they have already decided to convey.

Before continuing, we should explain a few terms. A *response option* describes a specific communicative act that might be performed as part of a response. Given a question like “is P true,” one response option might be “inform the questioner of P's truth value.” A respondent might choose to *exercise* a response option, meaning that she decides to perform the act described by the response option in her response. Exercising the option “inform the questioner of P's truth value” might lead to the appearance of “P is true” in a natural language response, assuming P were in fact true. Thus, response options are goals that a respondent wants to achieve in her response. The actual response is produced by forming and executing a *response plan* that achieves the goals specified by the response options.

The main arguments in this section will make use of two examples shown in Figure 1. These examples were extracted from transcripts of electronic mail conversations between consultants and users of the Unix¹ operating system at the University of California at Berkeley (henceforth the “Berkeley transcripts”).

2.1 Selecting among multiple response options

We begin by observing that the utterances comprising a query may license various response options, only some of which are exercised in any actual response. In Example 1, Q² has requested to be informed *if* a network link exists between his computer and ucb-euler. There are several pieces of information which R might consider conveying in her response to this question. She could certainly give the *direct answer*, telling

¹Unix is a trademark of AT&T.

²Throughout this paper, ‘Q’ (or a masculine pronoun) will denote the questioner, ‘R’ (or a feminine pronoun) the respondent.

Example 1

- Q:** Do you know if there is any network link to ucb-euler? I want to send some mail over there.
- R:** Gary, just do this: mail ruby!euler!(login name). Let us know if it doesn't work. Euler is only reached thru the ruby machine.

Example 2

- Q:** Is there a way to send mail to ucb-euler from ucb-cory?
- R:** Yes, it is letter y on the Berknet. So mail user@y.CC. If you have further problems with it, mail to serge@cory. He is the euler system manager.

Figure 1: Two Cooperative Exchanges

Q that there is in fact a network link between the two machines. In an effort to be helpful, she might tell Q *how many* such links there are, assuming a count of this kind is meaningful. Or she might tell Q *what kind* of links there are, e.g., a high-speed ethernet link, a low-speed phonenet connection, and so forth. Recast as response options, R might identify her options as “inform Q that a network link exists,” “inform Q of the count of network links,” and “inform Q of the type of each network link.”

R's possible response options follow from general principles of interaction and reasoning, based on beliefs about Q's goals and intended actions. On general principle, for example, queries that ask “whether P” can always be answered by informing Q of P's truth value. In response to “is there a P,” one might include a count of the Ps if there is more than one—“over-answering” the question, in the sense of Wahlster [15]. Reasoning on her beliefs about Q's plan, R may be able to identify potential *obstacles* [1] that can be eliminated by providing Q with some information.

In Example 1, R does not produce all possible response options. In fact, she does not even explicitly give the direct answer. The point is that the direct answer can be deduced from the response: there clearly is some network connection between the machines, at least one that permits the transmittal of mail messages. So either R has decided to convey the direct answer implicitly, or she has decided to leave it out, with the direct answer being implied purely by happenstance.

We take this as evidence that R considered her plausible response options and decided which to include in the final response. In terms of a computational model, R first identifies the available response options and then decides which ones to actually exercise.

There are different bases for rejecting (deciding not to exercise) a response option. For example, exercising one response option may make another unnecessary—replying “there are three Ps” when asked “is there a P” makes the direct answer of “yes” unnecessary. Alternatively, one response option may, in the given circumstance, be considered more important than another: correcting a misconception evident from Q's query may be more important than answering his question. Finally, a respondent may have to reject response options simply to avoid a lengthy reply.

The Berkeley transcripts contain many examples of open-ended queries in which Q primarily describes his goal, leaving it up to R to decide how to respond:

- Q:** I'm using a hardcopy printing terminal on a dialup line. In ex³ I use the space bar to indent lines in a program I'm entering. After I indent on one line, the next line automatically indents the same number of spaces, but I don't want it to. I got out of ex and then went back in and the indent was still set where it had been. Logging out removed the automatic indenting, but that's a hard way to work! Any suggestions?

³In Unix, “ex” refers to a line-oriented text editor.

Q's query ("any suggestions?") is a general request for help with the described problem. Here is another situation in which, in choosing her response, R may have to weigh different options. For example, she might recommend that Q switch to a different editor, or if automatic indentation indicates an improperly configured terminal, R might want to point that out.

Based on these arguments, we propose that a rating/selection phase be part of a computational model of cooperative response generation. The system should first collect all the response options that it deems are potentially relevant and useful. These options are then rated according to the kinds of criteria we mentioned earlier. The ratings then influence which options are chosen to be exercised in the system's cooperative response.

We want to emphasize that we believe rating/selection is a necessary attribute of CRG *systems*, not necessarily part of a model of human cooperative activity. Given a question, NLQA systems are not constrained as much as people are to begin responding immediately. They should be more circumspect in their choice of response options, and take more care to be brief yet informative, not to mislead, and to appropriately justify their assertions or directives. The rating/selection process enables CRG systems to be as cooperative as they can and need to be.

2.2 Reflecting on earlier decisions

We base our argument for including reflection in a computational model of CRG on Example 2 (Figure 1). Notice that R asserts:

He [serge@cory] is the euler system manager.

What might have motivated R to make this statement? There doesn't seem to be anything in Q's request, explicit or implicit, that suggests a need to be told anything about serge@cory or about euler's system manager. To account for this phenomenon, we start by examining the immediately preceding statement:

If you have further problems with it, mail to serge@cory.

Why might R have included this statement in her response? Since we are trying to develop a computational model of R's response, a better question is: what process could explain why R made the above statement part of her response?

Our analysis goes as follows: first, we can assume that R inferred Q's goal of sending mail to ucb-euler because the first part of her response helps Q reach that goal, by informing him of the correct electronic mail address syntax. That is, she chose to exercise the response option "inform Q how to send mail to ucb-euler." R's next statement, "if you have further problems..." is a further attempt to help Q send mail successfully to euler. Several explanations for its appearance exist, including:

1. R believes that Q may still have problems (not involving incorrect address syntax) sending mail to euler, and therefore needs to know of a better way (better than asking R) to handle those problems.
2. R is unsure that user@y.CC is in fact the correct address. However, she knows that serge@cory definitely knows the right way to send mail, so she points Q to him.
3. As a matter of policy, R routinely directs users to those people with the most expertise in the given problem area.

All explanations suggest that R's second utterance is still part of her effort to help Q reach his goal. She recognizes his goal, identifies what she takes to be his mistake (he was using the wrong address syntax), corrects his mistake, and then, allowing that other difficulties may arise, she points him to a better source of information. The presence of her third (last) utterance, though, seems to have a different explanation.

Looking at its effect on Q, "serge@cory is the euler system manager" explains R's second statement which mentions serge@cory for the first time. That is, having decided to refer to serge@cory, R realizes that she should also explain who he is.

This process we call *reflection*, to capture the idea that after selecting an initial set of response options to exercise, a respondent “reviews” or “reflects” on those choices and may be able to identify new response options that are suddenly relevant, relevant by dint of the information to be conveyed by response options already chosen. So some response options are chosen because they address the questioner’s goals, plans and needs, while other response options are selected to justify, clarify, or explain options that the respondent has already decided to exercise.

Through reflection, a respondent also may decide to generate a new plan for the questioner and communicate the important details to him. Consider this example:

Q: The Linc Lab laserwriter is down again. What’s Ira’s office phone number?

R: Ira’s not in his office. Call Dawn at extension 3191. She may be able to fix the problem, or can page Ira if necessary.

After inferring Q’s plan, R evaluates⁴ it and discovers an error: Q believes that by calling Ira’s office, he will contact Ira. R then decides to point that error out. Upon reflection, R notices that she has not helped Q reach his goal, so she decides to try to find another plan that Q could execute in order to get the Linc Lab laserwriter fixed. She finds such a plan, and decides to communicate the important details—that Q should contact Dawn at extension 3191.

Note that the example also shows that reflection make occur over several cycles before a final response is determined. R’s final statement in the example explains to Q why he should call Dawn—it explains the plan that R has decided to communicate. Computationally, reflecting upon her decision to tell Q to call Dawn, R decides that a justification of that plan is necessary.

Reflection is an important part of the cooperative response generation process. It allows CRG systems not only to explain, justify, or clarify their statements, but also allows them to perform other kinds of helpful actions such as suggesting new plans.

3 Concluding Remarks

We have proposed two characteristics of a computational model of cooperative response generation. The ideas described in this paper are embodied in a prototype CRG system that is being designed and implemented at the University of Pennsylvania. For more details on the motivation for and architecture of that system, see Cheikes [3].

References

- [1] James F. Allen. Recognizing intentions from natural language utterances. In Michael Brady and Robert C. Berwick, editors, *Computational Models of Discourse*, pages 107–166, The MIT Press, Cambridge, MA, 1983.
- [2] Sandra Carberry. Plan recognition and its use in understanding dialogue. In Alfred Kobsa and Wolfgang Wahlster, editors, *User Models in Dialog Systems*, Springer Verlag, Berlin–New York, 1988.
- [3] Brant A. Cheikes. *The Architecture of a Cooperative Respondent*. Technical Report MS-CIS-89-13, Department of Computer and Information Science, University of Pennsylvania, 1989.
- [4] Annie Gal. A natural language database interface that provides cooperative answers. In *Proceedings of the Second Conference on Artificial Intelligence Applications*, pages 352–357, 1985.
- [5] Julia B. Hirschberg. *A Theory of Scalar Implicature*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, December 1985.

⁴ *Plan evaluation* is discussed at length in Cheikes [3]. Briefly, plan evaluation is the process by which a respondent identifies errors in an inferred plan.

- [6] Aravind K. Joshi, Bonnie L. Webber, and Ralph Weischedel. Living up to expectations: computing expert responses. In *Proceedings of the 4th National Conference on Artificial Intelligence*, August 1984.
- [7] S. Jerrold Kaplan. Cooperative responses from a portable natural language database query system. In Michael Brady, editor, *Computational Models of Discourse*, The MIT Press, Cambridge, MA, 1982.
- [8] Eric Mays. Failures in natural language systems: application to data base query systems. In *Proceedings of the 1st National Conference on Artificial Intelligence*, Stanford, August 1980.
- [9] Eric Mays. A temporal logic for reasoning about changing data bases in the context of natural language question-answering. In Lawrence Kerschberg, editor, *Expert Database Systems*, Benjamin Cummings, New York, 1985.
- [10] Kathleen F. McCoy. *Correcting Object-Related Misconceptions*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, December 1985.
- [11] Kathleen R. McKeown. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, Cambridge, 1985.
- [12] Martha E. Pollack. A model of plan inference that distinguishes between actor's and observer's beliefs. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, New York, June 1986.
- [13] Alex Quilici, Michael Dyer, and Margot Flowers. Detecting and responding to plan-oriented misconceptions. In Alfred Kobsa and Wolfgang Wahlster, editors, *User Models in Dialog Systems*, Springer Verlag, Berlin-New York, 1988.
- [14] Lucy A. Suchman. *Plans and Situated Actions*. Cambridge University Press, Cambridge, 1987.
- [15] Wolfgang Wahlster, Heinz Marburger, Anthony Jameson, and Stephan Busemann. Over-answering yes-no questions: extended responses in a NL interface to a vision system. In *Proceedings of the 8th International Conference on Artificial Intelligence*, pages 643-646, Karlsruhe, August 1983.
- [16] Bonnie L. Webber. Questions, answers, and responses: a guide for knowledge based systems. In M. Brodie and J. Mylopoulos, editors, *On Knowledge Base Systems*, Springer-Verlag, Amsterdam, 1986.
- [17] Robert Wilensky, James Mayfield, Anthony Albert, David Chin, Charles Cox, Marc Luria, James Martin, and Dekai Wu. *UC: A Progress Report*. Technical Report UCB/CSD 87/303, Computer Science Division, University of California, Berkeley, July 1986.