University of Pennsylvania
ScholarlyCommons

Technical Reports (CIS)
Department of Computer & Information Science

1-1-2007

Living Up to Expectations: Computing Expert Responses

Aravind K. Joshi
University of Pennsylvania, joshi@cis.upenn.edu

Bonnie L. Webber
University of Pennsylvania, bonnie@inf.ed.ac.uk

Ralph Weischedel
University of Delaware

Follow this and additional works at: http://repository.upenn.edu/cis_reports

Recommended Citation

Aravind K. Joshi, Bonnie L. Webber, and Ralph Weischedel, "Living Up to Expectations: Computing Expert Responses", . January 2007.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-84-60.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/cis_reports/669
For more information, please contact libraryrepository@pobox.upenn.edu.

Living Up to Expectations: Computing Expert Responses

Abstract

In cooperative man-machine interaction, it is necessary *but not sufficient* for a system to respond truthfully and informatively to a user's question. In particular, if the system has reason to believe that its planned response might mislead the user, then it must block that conclusion by modifying its response. This paper focuses on identifying and avoiding potentially misleading responses by acknowledging types of 'informing behavior' usually expected of an expert. We attempt to give a formal account of several types of assertions that should be included in response to questions concerning the achievement of some goal (in addition to the simple answer), lest the questioner otherwise be misled.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-84-60.

This technical report is available at ScholarlyCommons: http://repository.upenn.edu/cis_reports/669

LIVING UP TO EXPECTATIONS:
COMPUTING EXPERT RESPONSES

Aravind Joshi
Bonnie Webber
Ralph Weischedel

MS-CIS-84-60

Department Of Computer and Information Science
Moore School
University of Pennsylvania
Philadelphia, PA 19104

1984

Acknowledgements: This research was supported in part by DARPA grants NOOO14-85-K-0018 and NOOO14-85-K-0807, NSF grants MCS8219196-CER, MCS-82-07294, 1 RO1-HL-29985-01, U.S. Army grants DAA6-29-84-K-0061, DAAB07-84-K-F077, U.S. Air Force grant 82-NM-299, AI Center grants NSF-MCS-83-05221, U.S. Army Research office grant ARO-DAA29-84-9-0027, Lord Corporation, RCA

LIVING UP TO EXPECTATIONS: COMPUTING EXPERT RESPONSES[1]

Aravind Joshi and Bonnie Webber
Department of Computer and Information Science
Moore School/D2
University of Pennsylvania
Philadelphia PA 19104

Ralph M. Weischedel[2]
Department of Computer & Information Sciences
University of Delaware
Newark DE 19716

ABSTRACT

In cooperative man-machine interaction, it is necessary *but not sufficient* for a system to respond truthfully and informatively to a user's question. In particular, if the system has reason to believe that its planned response might mislead the user, then it must block that conclusion by modifying its response. This paper focusses on identifying and avoiding potentially misleading responses by acknowledging types of "informing behavior" usually expected of an expert. We attempt to give a formal account of several types of assertions that should be included in response to questions concerning the achievement of some goal (in addition to the simple answer), lest the questioner otherwise be misled.

1. Introduction]

In cooperative man-machine interaction, it is necessary *but not sufficient* for a system to respond truthfully and informatively to a user's question. In particular, if the system has reason to believe that its planned response might mislead the user to draw a false conclusion, then it must block that conclusion by modifying or adding to its response.

Such cooperative behavior was investigated in [5], in which a modification of Grice's *Maxim of Quality* - "Be truthful" - is proposed:

> If you, the speaker, plan to say anything which may imply for the hearer something that you believe to be false, then provide further information to block it.

This behavior was studied in the context of interpreting certain definite noun phrases. In this paper, we investigate this revised principle as applied to responding to users' plan-related questions. Our overall aim is to:

1. characterize tractable cases in which the system as respondent (R) can anticipate the possibility of the user/questioner (Q) drawing false conclusions from its response and hence alter it so as to prevent this happening;
2. develop a formal method for computing the projected inferences that Q may draw from a

---

[1] This work is partially supported by NSF Grants MCS 81-07290, MCS 83-05221, and IST 83-11400.

[2] At present visiting the Department of Computer and Information Science, University of Pennsylvania PA 19104.

particular response, identifying those factors whose presence or absence catalyzes the inferences;

3. enable the system to generate modifications of its response that can defuse possible false inferences and that may provide additional useful information as well.

In responding to any question, including those related to plans, a respondent (R) must conform to Grice's first *Maxim of Quantity* as well as the revised *Maxim of Quality* stated above:

> Make your contribution as informative as is required (for the current purposes of the exchange).

At best, if R's response is not so informative, it may be seen as uncooperative. At worst, it may end up violating the revised *Maxim of Quality*, causing Q to conclude something R either believes to be false or does not know to be true: the consequences could be dreadful. Our task is to characterize more precisely what this expected informativeness consists of. In question answering, there seem to be several quite different types of information, over and beyond the simple answer to a question, that are nevertheless expected. For example,

1. When a task-related question is posed to an expert (R), R is expected to provide additional information that he recognizes as necessary to the performance of the task, of which the questioner (Q) may be unaware. Such response behavior was discussed and implemented by Allen [1] in a system to simulate a train information booth attendant responding to requests for schedule and track information. In this case, not providing the expected additional information is simply uncooperative: Q won't conclude the train doesn't depart at any time if R fails to volunteer one.

2. With respect to discussions and/or arguments, a speaker contradicting another is expected to support his contrary contention. Again, failing to provide support would simply be viewed as uncooperative [2, 3].

3. With respect to an expert's responses to questions, if Q expects that R would inform him of P if P were true, then Q may interpret R's silence regarding P as implying P is not true.[3] Thus if R knows P to be true, his silence may lead to Q's being misled. This third type of expected informativeness is the basis for the potentially misleading responses that we are trying to avoid and that constitute the subject of this paper.

What is of interest to us is *characterizing* the Ps that Q would expect an expert R to inform him of, if they hold. Notice that these Ps differ from script-based expectations [8], which are based on what is taken to be the ordinary course of events in a situation. In describing such a situation, if the speaker doesn't explicitly reference some element P of the script, the listener simply assumes it is true. On the other hand, the Ps of interest here are based on normal cooperative discourse behavior, as set out in Grice's maxims. If the speaker doesn't make explicit some information P that the listener believes he would possess and inform the listener of, the listener assumes it is false.

In this paper, we attempt to give a formal account of a subclass of Ps that should be included (in addition to the simple answer) in response to questions involving Q's achieving some goal[4] - e.g., "Can I drop CIS577?", "I want to enrol in CIS577?", "How do I get to Marsh Creek on the Expressway?", etc., lest that response otherwise mislead Q. In this endeavor, our first step is to specify that knowledge that an expert R must have in order to identify the Ps that Q would expect to be informed of, in response to his question. Our second step is to formalize that knowledge and show how the system can use it. Our third step is to show how the system can modify its planned response so as to convey those Ps. In this paper, Section 2 addresses the first step of this process and Sections 3 and 4 address the second. The third step we mention here only in passing.

---

[3] This is an interactional version of what Reiter [13] has called the "Closed World Assumption" and what McCarthy [9] has discussed in the context of "Circumscription".

[4] A companion paper [6] discusses responses which may mislead Q into assuming some default which R knows not to hold. Related work [4] discusses providing indirect or modified responses to yes/no questions where a direct response, while truthful, might mislead Q.

2. Factors in Computing Likely Informing Behavior]

Before discussing the factors involved in computing this desired system behavior, we want to call attention to the distinction we are drawing between *actions* and *events*, and between the *stated goal* of a question and its *intended goal*. We limit the term *action* to things that Q has some control over. Things beyond Q's control we will call *events*, even if performed by other agents. While events may be *likely* or even *necessary*, Q and R nevertheless can do nothing more than wait for them to happen. This distinction between actions and events shows up in R's response behavior: if an action is needed, R can suggest that Q perform it. If an event is, R can do no more than inform Q.

Our second distinction is between the *stated goal* or "S-goal" of a request and its *intended goal* or "I-goal". The former is the goal most directly associated with Q's request, beyond that Q know the information. That is, we take the S-goal of a request to be the goal directly achieved by using the information.

Underlying the stated goal of a request though may be another goal that the speaker wants to achieve. This *intended goal* or "I-goal" may be related to the S-goal of the request in any of a number of ways:

- The I-goal may be the same as the S-goal.
- The I-goal may be more abstract than the S-goal, which addresses only part of the I-goal. (This is the standard goal/sub-goal relation found in hierarchical planning [14].) For example, Q's S-goal may be to delete some files (e.g., "How can I delete all but the last version of FOO.MSS?"), while his I-goal may be to bring his file usage under quota. This more abstract goal may also involve archiving some other files, moving some into another person's directory, etc.
- The S-goal may be an enabling condition for the I-goal. For example, Q's S-goal may be to get read/write access to a file, while his I-goal may be to alter it.
- The I-goal may be more general than the S-goal. For example, Q's S-goal may be to know how to repeat a control-N, while his I-goal may be to know how to effect multiple sequential instances of a control character.
- Conversely, the I-goal may be more specific than the S-goal - for example, Q's S-goal may be to know how to send files to someone on another machine, while his I-goal is just to send a particular file to a local network user, which may allow for a specialized procedure.

Inferring the I-goal corresponding to an S-goal is an active area of research [1, Carberry83, 10, 11]. We assume for the purposes of this paper that R can successfully do so. One problem is that the relationship that Q believes to hold between his S-goal and his I-goal may not actually hold: for example, the S-goal may not fulfill part of the I-goal, or it may not instantiate it, or it may not be a pre-condition for it. In fact, the S-goal may not even be possible to effect! This failure, under the rubric "relaxing the appropriate-query assumption", is discussed in more detail in [10, 11]. It is also reason for augmenting R's response with appropriate Ps, as we note informally in this section and more formally in the next.

Having drawn these distinctions, we now claim that in order for the system to compute both a direct answer to Q's request and such Ps as he would expect to be informed of, were they true, the system must be able to draw upon knowledge/beliefs about

- the events or actions, if any, that can bring about a goal
- their enabling conditions
- the likelihood of an event occuring or the enabling conditions for an action holding, with respect to a state
- ways of evaluating methods of achieving goals - for example, with respect to simplicity, other consequences (side effects), likelihood of success, etc.
- general characteristics of cooperative expert behavior

The roles played by these different types of knowledge (as well as specific examples of them) are well illustrated in the next section.

3. Formalizing Knowledge for Expert Response

In this section we give examples of how a formal model of user beliefs about cooperative expert behavior can be used to avoid misleading responses to task-related questions - in particular, what is a very representative set of questions, those of the form "How do I do X?". Although we use logic for the model because it is clear and precise, we are not proposing theorem proving as the means of computing cooperative behavior. In Section 4 we suggest a computational mechanism. The examples are from a domain of advising students and involve responding to the request "I want to drop CIS577". The set of individuals includes not only students, instructors, courses, etc. but also states. Since events and actions change states, we represent them as (possibly parameterized) functions from states to states. All terms corresponding to events or actions will be underlined. For these examples, the following notation is convenient:

Q	the user
R	the expert
Sc	the current state of the student
RB(P)	R believes proposition P
RBQB(P)	R believes that Q believes P
admissible(e(S))	event/action e can apply in state S
likely(a,S)	a is a likely event/action in state S
holds(P,S)	P, a proposition, is true in S
want(x,P)	x wants P to be true

To encode the preconditions and consequences of performing an action, we adopt an axiomatization of STRIPS operators due to [Chester83, 7, 15]. The preconditions on an action being applicable are encoded using "holds" and "admissible" (essentially defining "admissible"). Namely, if c1, ..., cn are preconditions on an action a,

$$\mathrm{holds}(c1,s)\ \&...\&\ \mathrm{holds}(cn,s) \Rightarrow \mathrm{admissible}(a(s))$$

a's immediate consequences p1, ..., pm can be stated as

$$\mathrm{admissible}(a(s)) \Rightarrow \mathrm{holds}(p1,\ a(s))\ \&\ ...\ \&\ \mathrm{holds}(pm,\ a(s))$$

A frame axiom states that only p1, ..., pm have changed.

$$\neg(p{=}p1)\ \&\ ...\ \&\ \neg(p{=}pm)\ \&\ \mathrm{holds}(p,s)3\ \&\ \mathrm{admissible}(a(s)) \Rightarrow \mathrm{holds},a(s))$$

In particular, we can state the preconditions and consequences of dropping CIS577. (h and n are variables, while C stands for CIS577.)

$$\mathrm{RB}(\mathrm{holds}(\mathrm{enrolled}(h,\ C,\ \mathrm{fall}),\ n)\ \&\ \mathrm{holds}(\mathrm{date}(n){<}\mathrm{Nov16},\ n)$$
$$\Rightarrow \mathrm{admissible}(\mathit{drop}(h,C)(n)))$$

$$\mathrm{RB}(\mathrm{admissible}(\mathit{drop}(h,C)(n)) \Rightarrow \mathrm{holds}(\neg\mathrm{enrolled}(h,C,\mathrm{fall}),\mathit{drop}(h,C)(n)))$$

$$\mathrm{RB}(\neg(p{=}\mathrm{enrolled}(h,C,\mathrm{fall}))\ \&\ \mathrm{admissible}(\mathit{drop}(h,C)(n))\ \&\ \mathrm{holds}(p,n)$$
$$\Rightarrow \mathrm{holds}(p,\mathit{drop}(h,C)(n)))$$

Of course, this only partially solves the frame problem, since there will be implications of p1, ..., pm in general. For instance, it is likely that one might have an axiom stating that one receives a grade in a course only if the individual is enrolled in the course.

Q's S-goal in dropping CIS577 is not being in the course. By a process of reasoning discussed in [10, 11], R may conclude that Q's likely intended goal (I-goal) is not failing it. That is, R may believe:

$$\mathrm{RBQB}(\mathrm{holds}(\neg\mathrm{fail}(Q,C),\ \mathit{drop}(Q,C)(Sc)))^{5}$$

$$\mathrm{RB}(\mathrm{want}(Q,\neg\mathrm{fail}(Q,C))\ )$$

What we claim is: (1) R must give a truthful response addressing at least Q's S-goal; (2) in addition, R may have to provide information in order not to mislead Q; and (3) R may give additional information to be cooperative in other ways. In the subsections below, we enumerate the cases that R must check in effecting (2). In each case, we give both a formal representation of the additional information to be conveyed and a possible English gloss. In that gloss, the part addressing Q's S-goal will appear in normal type, while the additional information will be underlined.

For each case, we give two formulae: a statement of R's beliefs about the current situation and an axiom stating R's beliefs about Q's expectations. Formulae of the first type have the form RB(P). Formulae of the second type relate such beliefs to performing an informing action. They involve a statement of the form

$$\mathrm{RB}[P] \Rightarrow \mathrm{likely}(i,\ Sc),$$

where i is an informing act. For example, if R believes there is a better way to achieve Q's goal, R is likely to inform Q of that better way. Since it is assumed that Q has this belief, we have

$$\mathrm{QB}(\ \mathrm{RB}[P] \Rightarrow \mathrm{likely}(i,\ Sc)\ ).$$

---

[5] It will also be the case that RBQB(admissible(*drop(Q,C)*(Sc))) if Q's asks "How can I drop CIS577?", but not if he asks "Can I drop CIS577?". In the latter case, Q must of course believe that it may be admissible, or why ask the question. In either case, R's subsequent behavior doesn't seem contingent on his beliefs about Q's beliefs about admissibility.

where we can equate "Q believes i is likely" with "Q expects i." Since R has no direct access to Q's beliefs, this must be embedded in R's model of Q's belief space. Therefore, the axioms have the form (modulo quantifier placement)

$$\mathrm{RBQB}(\ \mathrm{RB}[P] \Rightarrow \mathrm{likely}(i,\ \mathrm{Sc})\ ).$$

An informing act is meant to serve as a command to a natural language generator which selects appropriate lexical items, phrasing, etc. for a natural language utterance. Such an act has the form

inform-that(R,Q,P) R informs Q that P is true.

3.1. Failure of enabling conditions

Suppose that it is past the November 15th deadline or that the official records don't show Q enrolled in CIS577. Then the enabling conditions for dropping it are not met. That is, R believes Q's S-goal cannot be achieved from Sc.

[1] $\mathrm{RB}(\mathrm{want}(Q,\neg\mathrm{fail}(Q,C))\ \&\ \neg\mathrm{admissible}(\mathit{drop}(Q,C)(\mathrm{Sc})))$

Thus R initially plans to answer "You can't drop CIS577". Beyond this, there are two possibilities.

3.1.1. A way

If R knows another action b that would achieve Q's goals (cf. formula [2]), Q would expect to be informed about it. If not so informed, Q may mistakenly conclude that there is no other way. Formula [3] states this belief that R has about Q's expectations.

[2] $\mathrm{RB}((\exists b)\ [\mathrm{admissible}(b(\mathrm{Sc}))\ \&\ \mathrm{holds}(\neg\mathrm{fail}(Q,C),\ b(\mathrm{Sc}))])$

[3] $$\begin{aligned}&\mathrm{RBQB}(\mathrm{RB}[\mathrm{want}(Q,\neg\mathrm{fail}(Q,C))\ \&\ \neg\mathrm{admissible}(\mathit{drop}(Q,C)(\mathrm{Sc}))]\ \&\\ &\quad \mathrm{RB}[(\exists b)[\mathrm{admissible}(b(\mathrm{Sc}))\ \&\ \mathrm{holds}(\neg\mathrm{fail}(Q,C),b(\mathrm{Sc}))]]\\ &\quad \Rightarrow \mathrm{likely}(\mathit{inform\text{-}that}(R,\ Q,\\ &\qquad\qquad (\exists b)\ [\mathit{admissible}(b(Sc))\ \&\ \mathit{holds}(\neg \mathit{fail}(Q,C),b(Sc))\ \&\\ &\qquad\qquad\quad \mathit{can}(Q,b)]),\mathrm{Sc})])\end{aligned}$$

R's full response is therefore "You can't drop 577; *you can b.*" For instance, b could be changing status to auditor, which may be performed until December 1.

3.1.2. No way

If R doesn't know of any action or event that could achieve Q's goal (cf. [4]), Q would expect to be so informed. Formula [5] states this belief about Q's expectations.

[4] $\mathrm{RB}(\neg(\exists a)[\mathrm{admissible}(a(\mathrm{Sc}))\ \&\ \mathrm{holds}(\neg\mathrm{fail}(Q,C),a(\mathrm{Sc}))])$

[5] $$\begin{aligned}&\mathrm{RBQB}(\mathrm{RB}(\mathrm{want}(Q,\neg\mathrm{fail}(Q,C))\ \&\ \neg(\exists a)[\mathrm{admissible}(a(\mathrm{Sc}))\\ &\qquad\qquad \&\ \mathrm{holds}(\neg\mathrm{fail}(Q,C),\ a(\mathrm{Sc}))])\\ &\qquad \Rightarrow \mathrm{likely}(\mathit{inform\text{-}that}(R,\ Q,\ \neg(\exists a)[\mathit{admissible}(a(Sc))\\ &\qquad\qquad \&\ \mathit{holds}(\neg \mathit{fail}(Q,C),a(Sc))]),\mathrm{Sc}))\end{aligned}$$

To say only that Q cannot drop the course does not exhibit expert cooperative behavior, since Q would be uncertain as to whether R had considered other alternatives. Therefore, R's full response is "You can't drop 577; *there isn't anything you can do to prevent failing.*"

Notice that R's analysis of the situation may turn up additional information which a cooperative expert could provide that does not involve avoiding misleading Q. For instance, R could indicate enabling conditions that prevent there being a solution: suppose the request to drop the course is made after the November 15th deadline. Then R would believe the following, in addition to [1]

$$\text{RB}(\text{holds}(\text{enrolled}(Q,C,\text{fall}),Sc)\ \&\ \text{holds}(\text{date}(Sc)>\text{Nov15},Sc))$$

More generally, we need a schema such as the following about Q's beliefs:

$$\begin{aligned}
&\text{RBQB}(\text{RB}[\text{want}(Q,\neg\text{fail}(Q,C))\\
&\qquad\quad \&\ (\text{holds}(P1,\ S)\ \&\ldots\&\ \text{holds}(Pn,\ S)\Rightarrow\text{admissible}(a(S)))\\
&\qquad\quad \&\ (\neg\text{holds}(Pi,\ S),\ \text{for some}\ \ Pi\ \text{above})]\\
&\qquad\Rightarrow\text{likely}(\mathit{inform\text{-}that}(R,Q,\neg \mathit{holds}(Pi,S)),S))
\end{aligned}$$

In this case the response should be "You can't drop 577; *Pi isn't true.*" Alternatively, the language generator might paraphrase the whole response as, "if Pi were true, you could drop."

Of course there are potentially many ways to try to achieve a goal: by a single action, by a single event, or by an event and an action, ... In fact, the search for a sequence of events or actions that would achieve the goal may consider many alternatives. If all fail, it is far from obvious which blocked condition to notify Q of, and knowledge is needed to guide the choice. Some heuristics for dealing with that problem are given in [12].

3.2. An nonproductive act

Suppose the proposed action does not achieve Q's I-goal, cf. [6]. For example, dropping the course may still mean that failing status would be recorded as a WF (withdrawal while failing). R may initially plan to answer "You can drop 577 by ...". However, Q would expect to be told that his proposed action does not achieve his I-goal. Formula [7] states R's belief about this expectation.

$$[6]\ \text{RB}(\neg\text{holds}(\neg\text{fail}(Q,C),\ \mathit{drop}(Q,C)(\text{Sc}))\ \&\ \text{admissible}(\mathit{drop}(Q,C)(\text{Sc}))\ )$$

$$\begin{aligned}
&[7]\ \text{RBQB}(\text{RB}[\ \text{want}(Q,\neg\text{fail}(Q,C))\ \&\ \neg\text{holds}(\neg\text{fail}(Q,C),\mathit{drop}(Q,C)(\text{Sc}))\\
&\qquad\qquad\quad \&\ \text{admissible}(\mathit{drop}(Q,C)(\text{Sc}))]\\
&\qquad\quad\Rightarrow\text{likely}(\mathit{inform\text{-}that}(R,Q,\\
&\qquad\qquad\quad \neg \mathit{holds}(\neg \mathit{fail}(Q,C),\mathit{drop}(Q,C)(Sc))),\text{Sc}))
\end{aligned}$$

R's full response is, "You can drop 577 by .... *However, you will still fail.*" Furthermore, given the reasoning in section 3.1.1 above, R's full response would also inform Q if there is an action *b* that the user can take instead.

3.3. A better way

Suppose R believes that there is a better way to achieve Q's I-goal, cf. [8] - for example, taking an incomplete to have additional time to perform the work, and thereby not losing all the effort Q has already expended. Q would expect that R, as a cooperative expert, would inform him of such a better way, cf. [9]. If R doesn't, R risks misleading Q that there isn't one.

$$\begin{aligned}
&[8]\ \text{RB}((\exists b)[\text{holds}(\neg\text{fail}(Q,C),\ b(\text{Sc}))\ \&\\
&\qquad\quad \text{admissible}(b(\text{Sc}))\ \&\ \text{better}(b,\mathit{drop}(Q,C)(Sc))])
\end{aligned}$$

$$\begin{aligned}
&[9]\ \text{RBQB}(\text{RB}[\text{want}(Q,\neg\text{fail}(Q,C))]\ \&\\
&\qquad\qquad \text{RB}[(\exists b)[\text{holds}(\neg\text{fail}(Q,C),\ b(\text{Sc}))\ \&\ \text{admissible}(b(\text{Sc}))\ \&\\
&\qquad\qquad\quad \text{better}(b,\mathit{drop}(Q,C)(Sc))\\
&\qquad\Rightarrow\text{likely}(\mathit{inform\text{-}that}(R,Q,
\end{aligned}$$

$$\begin{aligned}&(\exists b)[\mathit{holds}(\neg \mathit{fail}(Q,C),b(Sc)) \mathbin{\&} \mathit{admissible}(b(Sc)) \mathbin{\&}\\ &\quad \mathit{better}(b,\mathit{drop}(Q,C)(Sc))],\ \mathrm{Sc})])\end{aligned}$$

R's direct response is to indicate how f can be done. R's full response includes, in addition, "*b is a better way.*"

Notice that if R doesn't explicitly tell Q that he is presenting a better way (i.e., he just presents the method), Q may be misled that the response addresses his S-goal: i.e., he may falsely conclude that he is being told how to drop the course. (The possibility shows up clearer in other examples - e.g., if R omits the first sentence of the response below

> Q: How do I get to Marsh Creek on the Expressway?
> R: It's faster and shorter to take Route 30. Go out
> Lancaster Ave until....

Thus even when adhering to expert response behavior in terms of addressing an I-goal, we must keep the system aware of potentially misleading aspects of its modified response as well.

Note that R may believe that Q expects to be told the best way. This would change the second axiom to include within the scope of the existential quantifier

$$(\forall a)\{\neg(a{=}b) \Rightarrow [\mathrm{holds}(\neg\mathrm{fail}(Q,C),\ a(\mathrm{Sc})) \mathbin{\&} \mathrm{admissible}(a(\mathrm{Sc})) \mathbin{\&} \mathrm{better}(b,a)]\}$$

3.4. The only way

Suppose there is nothing inconsistent about what the user has proposed - i.e., all preconditions are met and it will achieve the user's goal. R's direct response would simply be to tell Q how. However, if R notices that that is the only way to achieve the goal (cf. [10]), it could optionally notify Q of that, cf. [11].

[10] $\mathrm{RB}((\exists! a)[\mathrm{holds}(\neg\mathrm{fail}(Q,C),a(\mathrm{Sc})) \mathbin{\&} \mathrm{admissible}(a(\mathrm{Sc})) \mathbin{\&} a{=}\mathit{drop}(Q,C)(Sc)])$

[11] $$\begin{aligned}&\mathrm{RBQB}(\mathrm{RB}(\mathrm{want}(Q,\neg\mathrm{fail}(Q,C)))\\ &\qquad \mathbin{\&} \mathrm{RB}((\exists! a)[\mathrm{holds}(\neg\mathrm{fail}(Q,C),\ a(\mathrm{Sc})) \mathbin{\&} \mathrm{admissible}(a(\mathrm{Sc})) \mathbin{\&} a{=}\mathit{drop}(Q,C)(Sc)])\\ &\quad \Rightarrow \mathrm{likely}(\mathit{inform\text{-}that}(R,\ Q,\\ &\qquad\qquad (\exists! a)[\mathit{holds}(\neg \mathit{fail}(Q,C),a(Sc))\\ &\qquad\qquad\quad \mathbin{\&} \mathit{admissible}(a(Sc)) \mathbin{\&} a{=}\mathit{drop}(Q,C)(Sc)]),\ \mathrm{Sc}))\end{aligned}$$

R's full response is "You can drop 577 by .... *That is the only way to prevent failing.*"

3.5. Something Turning Up

Suppose there is no appropriate action that Q can take to achieve his I-goal. That is,

$$\mathrm{RB}(\ \neg(\exists\ a)[\mathrm{admissible}(a(\mathrm{Sc})) \mathbin{\&} \mathrm{holds}(g,\ a(\mathrm{Sc}))])$$

There may still be some event e out of Q's control that could bring about the intended goal. This gives several more cases of R's modifying his response.

3.5.1. Unlikely event

If e is unlikely to occur (cf. [12]), Q would expect R to inform him of e, while noting its implausibility, cf. [13]

[12] $$\begin{aligned}&\mathrm{RB}((\exists e)[\mathrm{admissible}(e(\mathrm{Sc})) \mathbin{\&} \mathrm{holds}(\neg\mathrm{fail}(Q,C),\ e(\mathrm{Sc}))\\ &\qquad \mathbin{\&} \neg\mathrm{likely}(e,\ \mathrm{Sc})])\end{aligned}$$

$$
\begin{aligned}
&[13]\ \mathrm{RBQB}(\mathrm{RB}(\mathrm{want}(Q,\neg\mathrm{fail}(Q,C))\ \&\\
&\qquad \mathrm{RB}(\neg(\exists a)[\mathrm{admissible}(a(\mathrm{Sc}))\ \&\ \mathrm{holds}(\neg\mathrm{fail}(Q,C),a(\mathrm{Sc}))]\ \&\\
&\qquad\qquad (\exists e)[\mathrm{admissible}(e(\mathrm{Sc}))\ \&\ \mathrm{holds}(\neg\mathrm{fail}(Q,C),e(\mathrm{Sc}))\\
&\qquad\qquad\qquad \&\ \neg\mathrm{likely}(e,\mathrm{Sc})])\\
&\qquad \Rightarrow \mathrm{likely}(\mathit{inform\text{-}that}(R,Q,\\
&\qquad\qquad (\exists e)[\mathit{admissible}(e,Sc)\ \&\ \mathit{holds}(\neg\mathit{fail}(Q,C),e(Sc))\\
&\qquad\qquad\qquad \&\ \neg\mathit{likely}(e,Sc)]),\mathrm{Sc}))
\end{aligned}
$$

Thus R's full response is, "You can't drop 577. *If e occurs, you will not fail 577, but e is unlikely.*"

3.5.2. Likely event

If the event *e* is likely (cf. [14]), it does not seem necessary to state it, but it is certainly safe to do so. A formula representing this case follows.

$$
\begin{aligned}
&[14]\ \mathrm{RB}((\exists e)[\mathrm{admissible}(e(\mathrm{Sc}))\ \&\\
&\qquad \mathrm{holds}(\neg\mathrm{fail}(Q,C),e(\mathrm{Sc}))\ \&\ \mathrm{likely}(e,\mathrm{Sc})])
\end{aligned}
$$

R's beliefs about Q's expectations are the same as the previous case except that likely(e, Sc) replaces ¬likely(e, Sc). Thus R's full response may be "You can't drop 577. *However, e is likely to occur, in which case you will not fail 577.*"

3.5.3. Event followed by action

If event *e* brings about a state in which the enabling conditions of an effective action *a* are true, cf. [15]

$$
\begin{aligned}
&[15]\ \mathrm{RB}((\exists e)(\exists a)[\mathrm{admissible}(e(\mathrm{Sc}))\ \&\ \mathrm{admissible}(a(e(\mathrm{Sc})))\ \&\\
&\qquad\qquad \mathrm{holds}(\neg\mathrm{fail}(Q,C),\ a(e(\mathrm{Sc})))])
\end{aligned}
$$

$$
\begin{aligned}
&[16]\ \mathrm{RBQB}(\mathrm{RB}((\exists e)(\exists a)[\mathrm{want}(Q,\neg\mathrm{fail}(Q,C))\ \&\ \mathrm{admissible}(e(\mathrm{Sc}))\\
&\qquad\qquad \&\ \mathrm{admissible}(a(e(\mathrm{Sc})))\ \&\ \mathrm{holds}(\neg\mathrm{fail}(Q,C),a(e(\mathrm{Sc})))])\\
&\qquad \Rightarrow \mathrm{likely}(\mathit{inform\text{-}that}(R,Q,\\
&\qquad\qquad (\exists e)(\exists a)\ [\mathit{holds}(\neg\mathit{fail}(Q,C),a(e(Sc))))\ \&\\
&\qquad\qquad\quad \mathit{admissible}(a(e(Sc)))])),\mathrm{Sc}))
\end{aligned}
$$

then the same principles about informing Q of the likelihood or unlikelihood of *e* apply as they did before. In addition, R must inform Q of *a*, cf. [16]. Thus R's full response would be "You can't drop 577. *If e were to occur, which is (un)likely, you could a and thus not fail 577.*"

4. Reasoning

Our intent in using logic has been to have a precise representation language whose syntax informs R's reasoning about Q's beliefs. Having computed a full response that conforms to all these expectations, R may go on to 'trim' it according to principles of brevity that we do not discuss here.

Our proposal is that the informing behavior is "pre-compiled". That is, R does not reason explicitly about Q's expectations, but rather has compiled the conditions into a case analysis similar to a discrimination net. For instance, we can represent informally several of the cases in section 3.

```
if $\mathrm{admissible}(\mathit{drop}(Q,C)(\mathrm{Sc}))$
  then if $\neg\mathrm{holds}(\neg\mathrm{fail}(Q,C),\mathit{drop}(Q,C)(\mathrm{Sc}))$
     then begin nonproductive act
        if $(\exists b)[\mathrm{admissible}(b(\mathrm{Sc}))\ \&\ \mathrm{holds}(\neg\mathrm{fail}(Q,C),b(\mathrm{Sc}))]$
           then a way
           else no way
        end
     else if $(\exists b)[\mathrm{admissible}(b(\mathrm{Sc}))\ \&$
```

```
                    holds(¬fail(Q,C),b(Sc)) & better(b,f)]
          then a better way
  else if (∃ b)[admissible(b(Sc)) & holds(¬fail(Q,C), b(Sc))]
        then a way
        else no way
  ...
```

Note that we are assuming that R assumes the most demanding expectations by Q. Therefore, R can reason solely within its own space without missing things.

5. Conclusion

Since the behavior of expert systems will be interpreted in terms of the behavior users expect of cooperative human experts, we (as system designers) must understand such behavior patterns so as to implement them in our systems. If such systems are to be truly cooperative, it is not sufficient for them to be simply truthful. Additionally, they must be able to predict limited classes of false inferences that users might draw from dialogue with them and also to respond in a way to prevent those false inferences. The current enterprise is a small but non-trivial step in this direction. In addition to questions about achieving goals, we are investigating other cases where a cooperative expert should prevent false inferences by another agent, including preventing inappropriate default reasoning [6, JWW84nonmon].

Future work should include

- identification of additional cases where an expert must prevent false inferences by another agent,
- formal statement of a general principle for constaining the search for possible false inferences, and
- design of a natural language planning component to carry out the informing acts assumed in this paper.

ACKNOWLEDGEMENTS

We would like to thank Martha Pollack, Deborah Dahl, Julia Hirschberg, Kathy McCoy and the AAAI program committee reviewers for their comments on this paper.

References

⋮

1. Allen, J. Recognizing Intentions from Natural Language Utterances. In *Computational Models of Discourse*, M. Brady, Ed., MIT Press, Cambridge MA, 1982.

2. Birnbaum, L., Flowers, M. & McQuire, R. Towards an AI Model of Argumentation. Proceedings of 1980 Conference, American Assoc. for Artificial Intelligence, Stanford CA, August, 1980.

3. Cohen, R. A Theory of Discourse Coherence for Argument Understanding. Proceedings of the 1984 Conference, Canadian Society for Computational Studies of Intelligence, University of Western Ontario, London Ontario, May, 1984, pp. 6-10.

4. Hirschberg, J. Scalar Implicature and Indirect Responses in Question-Answering. Proc. CSCSI-84, London, Ontario, May, 1984.

5. Joshi, A.K. Mutual Beliefs in Question Answering Systems. In *Mutual Belief*, N. Smith, Ed., Academic Press, New York, 1982.

6. Joshi, A., Webber, B. & Weischedel, R. Preventing False Inferences. Proceedings of COLING-84, Stanford CA, July, 1984.

7. Kowalski, Robert. *Logic for Problem Solving*. North Holland, New York, 1979.

8. Lehnert, W. A Computational Theory of Human Question Answering. In *Elements of Discourse Understanding*, A. Joshi, B. Webber & I. Sag, Ed., Cambridge University Press, 1981.

9. McCarthy, John. "Circumscription -- A Form of Non-Monotonic Reasoning". *Artificial Intelligence 13* (1980), 27-39.

10. Pollack, Martha E. Goal Inference in Expert Systesm. MS-CIS-84-07, University of Pennsylvania, 1984. Doctoral dissertaion proposal..

11. Pollack, M. Good Answers to Bad Questions. Proc. Canadian Society for Computational Studies of Intelligence (CSCSI), Univ. of Western Ontario, Waterloo, Canada, May, 1984.

12. Ramshaw, Lance and Ralph M. Weischedel. Problem Localization Strategies for Pragmatics Processing in Natural Language Front Ends. Proceedings of COLING-84, July, 1984.

13. Reiter, R. Closed World Databases. In *Logic and Databases*, H. Gallaire & J. Minker, Ed., Plenum Press, 1978, pp. 149-177.

14. Sacerdoti, Earl D.. *A Structure for Plans and Behavior*. American Elsevier, New York, 1977.

15. Warren, D.H.D. WARPLAN: A System for Generating Plans. Proceedings of IJCAI-75, August, 1975.