12

In Response: Next Steps in Natural Language Interaction

Bonnie Lynn Webber and Tim Finin
DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
UNIVERSITY OF PENNSYLVANIA
PHILADELPHIA PA 19104

In the area of man-machine interaction, Natural Language has so far primarily been used to simplify people's access to information. The next step beyond simple data access is the kind of cooperative interactive problem-solving that current expect systems aspire to. But support for problem-solving (which includes helping the user formulate his/her problems) demands more in the way of interaction than just answering requests for factual information. In the first part of this paper, we illustrate some of these needed capabilities. In the remainder, we discuss two of them in greater detail: (a) recognizing and responding to user misconceptions and (b) getting from users the information needed to help them solve their problems.

1. Introduction

In the area of man-machine interaction, Natural Language has primarily been seen as a significant way of simplifying people's access to system services and information. Potential users need not spend time learning or trying to remember some arcane formalism: they can express their requests as they would everyday.

This viewpoint has led to valuable research on removing what appear to be arbitrary, artificial constraints on a user's freedom of expression: parsers can handle most of English syntax (Bobrow, 1978; Robinson, 1982); domain-specific processors can be tuned to interpret most reasonable utterances within the domain (Shwartz, 1984); utterances can be interpreted to some extent in the context of the previous discourse (Grosz, 1977; Hendrix, et al., 1978; Sidner, 1982; Waltz, 1978; Webber, 1978) and in light of the user's underlying intentions (Allen, 1982; Cohen, et al., 1981; Sidner, 1981); misspelling and grammatical errors can be tolerated to some extent (Hendrix, et al., 1978; Kwasny & Sondheimer, 1981); and in combined graphics/Natural Language systems, utterances can be combined with pointing for added naturalness (cf. Woods, 1984).

On the other hand, this viewpoint seems to assume that users know what information they want and can use it to solve their problems themselves. The former, unfortunately, is not always the case, and the latter is not always possible. In fact, the frequent lack of in-house expertise needed for solving problems is often cited as the reason for developing "expert systems" in the first place.

But support for problem solving (which includes helping the user **formulate** his/her problems) demands more in the way of interaction than just answering requests for factual information. In the first part of this paper, we illustrate some of these needed capabilities. In the remainder, we discuss two of them in greater detail to show the kind of system support they require.

Our point is that if we are to go beyond simple data access via Natural Language to the next step — cooperative problem-solving interactions — we must look at the system's role in the interaction. If we fail to recognize this complementary issue, much of the advantage of Natural Language input will be lost.

2. *Problem-Solving Interactions*

What we aspire to is the type of discourse behavior displayed in cooperative interactive problem solving among humans. To characterize such interactions, two of our students (Pollack, et al., 1982) have collected and analysed transcripts of a "naturally occurring" expert system — the radio talk show "Harry Gross: Speaking of Your Money" (WCAU, Philadelphia). In this program, listeners call in to ask for financial advice, which the expert, Harry Gross, attempts to provide. The ensuing discourse is basically a cooperative problem solving interaction. While not a perfect model for machine-based expert systems, the talk show transcripts do suggest many types of interactions that these systems should endeavor to support.

An examination of these transcripts reveals a regular pattern of interaction, rather like a negotiation — the process through which people arrive at a conclusion by means of a discussion. Rarely does a caller simply state a problem and passively listen to the expert's response. Rather what ensues is a collaborative dialogue in which caller and expert negotiate to determine a statement of the problem the caller *wants* solved and the expert *can* solve, and the statement of a solution the expert can support and the caller can accept and ideally understand.

More specifically, we have noticed such activities as the following during these negotiations. ('H' stands for the expert, Harry Gross, and 'U', for the caller.)

- The user attempts to verify his/her understanding of what the expert has said and the expert responds to either confirm or clarify — e.g.,

  H: Okay, in your case I would not object to see you go back into that for another 6 months.
  U: So you roll it over, in other words?
  H: Right.

- The user suggests an alternative solution to that proposed by the expert, and the expert responds either to confirm its possibility or to show why not — e.g.,

  H: Put the money aside in T-notes.
  U: Now wait. In a 43% bracket I didn't think that would be wise. I thought maybe we should buy municipal bonds.
  H: If you buy municipals, the interest on your loan won't be deductible. So municipals just don't make sense.

- The user requests justification of the expert's suggestion, and the expert provides it — e.g.,

  H: You can stop right there. Take the money.
  U: Take the money?
  H: Right. You're only getting $1500 a year. At $17,000, no trouble at all to get 10% on $17,000.

- The caller shows confusion about a term or concept or explicitly requests information on it, and the expert provides clarification — e.g.,

  H: I'd like to see you put that into two different Southern utilities.
  U: Southern utilities?
  H: Yes.
  U: Huh?
  H: Utilities that operate in the South — Texas, Oklahoma, Florida, Georgia, . . .

- The expert requests information from the user and then works with him/her to get it — e.g.,

H: Have you gains on other securities?
U: Yeh, we have some certificates and real estate.
H: Sorry, I'm not making myself clear — if you had some other stock
U: Oh
H: Do you have any paper gains? On other securities?

- The user asks or answers a question, states a preference, etc. that shows a misconception, and the expert attempts to correct it, instead of or in addition to responding to his/her utterance. (Examples of this activity will be given in section 3.)

While these and other activities are discussed in more detail in Pollack, et al., (1982), their relevance here is that if the system is not capable of reacting appropriately in such interactions, the user may become confused by what the system **does** do in response. For example, consider utterances of the form "What about <x>?". At least two major systems (Hendrix, et al., 1978; Waltz, 1978) treat all such utterances as a short way of asking a parallel question to one asked earlier — e.g.,

U: What is the length of each Russian aircraft carrier?
S: 420 feet
U: What about the draft?

The second question is correctly taken to mean "What is the draft of each Russian aircraft carrier?" But utterances of that form can in fact also be used for another purpose mentioned above — to propose an alternative possible answer to the one given by the system. For example,

U: What's a good thing to invest my pension in?
S: Put the money aside in T-notes.
U: What about municipal bonds?

It would be devastating to treat the user's second question as a request for a good thing in which to invest municipal bonds! The conclusion is that systems must be able to perform additional functions (such as considering the user's proposed alternative) and recognize when they are called for.

A growing number of researchers are involved in developing the capabilities needed for Natural Language problem-solving interactions. This work includes that of Swartout (1981) and Wallis and Shortliffe (1982) *explanation,* McKeown (1982) *term definition,* and Wahlster and his colleagues (van Hahn, et al., 1980), Wilensky (1982), Woods (1984), Schank and Slade (1984), and Shwartz (1984) *advisory system structure.* The interactional capabilities we will be discussing in the remainder of the paper are not covered in this other work: section 3 focusses on recognizing and responding to user misconceptions, and section 4, on getting from the user the information a system needs to help the user solve his/her problems.

What we hope to gain by this discussion, besides the reader's increased awareness of the importance of the system's role in the interaction, is recognition of the fact that in order to perform at this more sophisticated level, systems need **both** enrichments to existing data models or knowledge representations **and** additional types of reasoning. In other words, appropriate interactive behavior will not come about merely by tacking onto an existing system some off-the-shelf front end. The mechanisms that bring it about will have to tie deeply into what must already be rich and powerful representation and reasoning components.

We want to emphasize also that spontaneous automatic generation of fluent Natural Language, though a desirable goal, is not the primary point here. The interactional capabilities that we discuss are needed even if a system is accessed using a formal notation: database systems will have to be as careful not to mislead users by their responses (Webber, 1983), while even with canned text, expert systems will have to make similar provision for getting their questions answered.

What is important — and this we get by looking at Natural Language interactions as our model — is that an interactive system, whatever it is, must follow everyday conversational principles and practices. If it does, a user's **normal expectations** about responses to his/her utterances and **normal strategies** for interpreting those responses will not fail him/her, even though the conversational partner is a machine.

3. *Recognizing and Responding to Misconceptions*

Much of our knowledge of the world is incomplete; a lot of it is faulty. Much of the time, it makes no difference. At times though, it does — for example, when trying to solve a problem or acquire some information. At those times, misconceptions may lead one to try to solve the wrong problem, to seek an inappropriate solution or to misunderstand and hence be misled by the information one receives. It is the latter point that the work described here addresses. User misconceptions about the domain or its encoding in the database or expert system may lead the user to draw false conclusions from the system's response to his/her question. The system must do what it can to prevent this. We shall discuss four types of user misconceptions here: (a) misconceptions that something exists (or that the system knows of its existence), (b) misconceptions that something can participate in some relation, (c) misconceptions about the classification or properties of some object, and (d) misconceptions that some event can occur.

3.1. "*Extensional*" Misconceptions

As many people have noted, most database queries can be considered requests for the extension of some set descriptor (i.e., a listing of the individuals satisfying that description). Such descriptors are made up by restricting descriptions of larger sets in various ways. For example, the question

> Which foreign-born employees work in the shoe department?

can be taken as a request for the set of individuals satisfying the description "foreign-born employee who works in the shoe department". This is composed of the more inclusive descriptor "employee" restricted to those who are foreign-born, restricted again to those who work in the shoe department. One obvious misconception that a user can hold in asking such a question is that some description has a nonempty extension in the database, when in fact it doesn't. In that case, the answer to the user's question will follow trivially, without the user realizing it. If the system can instead recognize and point out such misconceptions, the user will be better off. This is the aim of the CO–OP system developed at the University of Pennsylvania in 1979 (Kaplan, 1982).

For example, a question like "Which French majors failed CIS531 last term?" reveals *inter alia* the questioner's belief that there are French majors. If there are in fact no French majors, an unqualified "None", while technically correct, would confirm the questioner's false belief. What are the consequences of unintentionally confirming such beliefs if they are incorrect? If the questioner concludes from "None" that no French majors failed CIS531 last term, s/he might in turn believe that all French majors **passed** CIS531 last term, and in turn many more unwarranted things about the abilities of French majors.

CO–OP detects such misconceptions in the course of retrieving answers to the database query viewed as a composite set descriptor. If one of the more inclusive subset descriptions making up the query is found to have an empty extension, then search halts and the user is informed about the system's lack of knowledge of individuals satisfying the failing descriptions — e.g., "According to this database, there are no French majors." The user can then take this information into account in formulating further queries.

3.2. "*Type*" Misconceptions

A second common type of misconception is that some entity or subset of entities can participate in some relation. This is similar to type violations in programming languages, where, for example, a function or procedure call may be incorrect because its arguments are of the wrong type. Some initial work in this area is reported in Mays (1980). The knowledge needed to recognize type failures in users' queries is contained in the system's database schema and consists of entity-relation information, hierarchical (subset-superset) information, as well as partition information as to what collection of subsets of a given set are mutually exclusive. It is the last factor that is critical for distinguishing between a nondeviant request like

Which women teach courses?

and a deviant one like

Which undergraduates teach courses?

where — as shown in Figure 1 — the TEACH relation is asserted to hold between FACULTY and COURSE. As Figure 1 also shows, the entity PEOPLE is partitioned in two different ways — into MEN and WOMEN, and into FACULTY and STUDENT. Thus if an entity is classified as a MAN, it cannot also be classified as a WOMAN. But it can also be classified as FACULTY or STUDENT (but not both). Assuming a relation is always asserted at the most general point in the hierarchy, the meaning of the configuration is taken to be that **only** FACULTY teach COURSEs. STUDENTs cannot TEACH COURSEs, nor can any subset of STUDENT do so. Hence the TEACH relationship cannot hold between UNDERGRADUATE and COURSE. The same

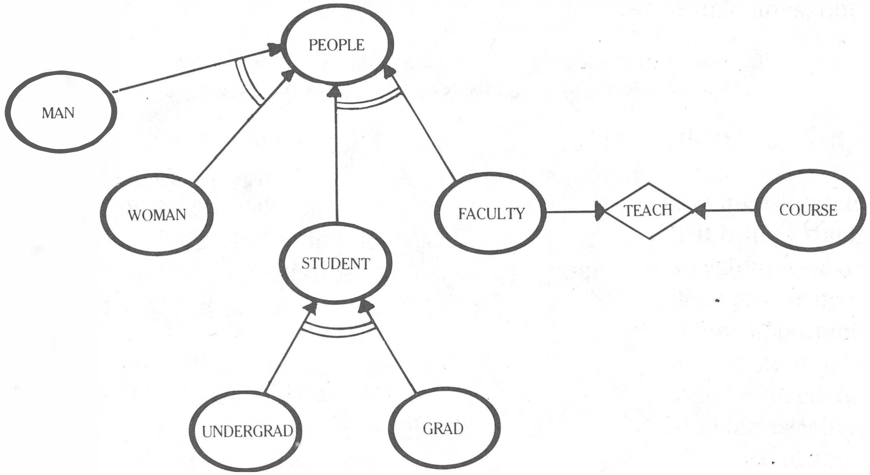

Figure 1. *Segment of database schema.*

is **not** true of WOMEN, as entities can be classified as both WOMEN and FACULTY.

In the case of type failures that it recognizes, Mays' system does not answer the user's query directly but rather provides relevant information about the entities and relation involved. That is, if the misconception involves a relation R holding between entities A and B, the system seeks relevant information by abstracting on each of R, A and B and determining whether either RA_ or R_B or _AB can hold in the database for some value of "_". For example, the system's response to the above question "Which undergraduates teach courses?" would be

> S: I don't believe that undergraduates can teach courses.
> Faculty **teach courses.**
> **Undergraduates** take **courses.**

Further work is needed on several problems involved in recognizing and responding to "type" misconceptions: first, in the context of negative questions — e.g.,

> U: Which courses are not taught by faculty?
> S: I believe that only faculty can teach courses. Therefore all courses are taught by faculty.

The user's question reveals an incorrent belief that the TEACH relation can hold of some entity *other than* FACULTY. May's system would not catch this misconception because it only tries to confirm the explicitly mentioned types — i.e., FACULTY, COURSE and TEACH. Further work is also needed on misconceptions involving "complex" types (requiring a more highly structured knowledge representation than Mays' system employed, for example Kl–One (Schmolze & Brachman, 1982)) — e.g.,

> U: How many widgets did we buy from Ajax last year?
> S: Ajax doesn't supply widgets — they supply clocks.

The relation BUY holds of a CUSTOMER, a SUPPLIER and a PRODUCT, and all these simple type constraints are satisfied in the question. However, there is in fact an additional constraint that the PRODUCT BOUGHT from a SUPPLIER is part of the PRODUCT LINE of that SUPPLIER. One would like to enable a system to apply this constraint both in standard updating procedures and as a way of recognizing and responding to "type" misconceptions.

3.3. "*Object-related*" *Misconceptions*

A third type of user misconception involves concepts modelled by the system. That is, there is some discrepancy between what the user believes about an object and what the system believes about it. This discrepancy may be revealed in a question the user asks the system — e.g.,

U: What is the interest rate on this stock?

or in his/her response to a question asked by the system — e.g.,

S: Do you have any liquid assets?
U: I have a $5K money market certificate.

As above, such discrepancies can either prevent the system from answering the user's query or prevent the user from correctly understanding the system's response or prevent the system from getting the information it needs (see also section 4). Thus the system should respond to the discrepancy as effectively as possible. Enabling it to do so in a domain-independent way is the aim of current research here being done by McCoy (1983).

When a person in conversation with another makes a error that may be the result of a misconception (as opposed to just a mistake), the very least the person expects from his/her partner is to have the false information denied — e.g.,

U: What is the interest rate on this stock?
S: Stocks don't have an interest rate.

S: Do you have any liquid assets?
U: I have a $5K money market certificate.
S: A money market certificate isn't a liquid asset.

However, through studying transcripts of people interacting with various experts (Malhotra, 1975; Malhotra & Sheridan, 1976; Pollack, et al., 1982; Schuster, 1982), McCoy has identified additional types of information often included in such responses — for example, a true statement that stands in contrast with the incorrect one — e.g.,

U: What is the interest rate on this stock?
S: Stocks don't have an interest rate. Stocks may pay a dividend.

or information aimed at convincing the user of the correction or at pointing out and refuting the faulty reasoning perceived to have been done — e.g.,

U: What is the interest rate on this stock?
S: Stocks don't have an interest rate. Only investments which have a face value (like bonds) have an interest rate.

S: Do you have any liquid assets?
U: I have a $5K money market certificate.
S: A money market certificate isn't a liquid asset. Your money is tied up for a period of several years in a money market certificate.

or even information that suggests a correction — e.g.,

> S: Do you have any liquid assets?
> U: I have a $5K money market certificate.
> S: A money market certificate isn't a liquid asset. Perhaps you meant a money market fund.

McCoy's problem is two-fold: (a) to characterize in a domain-independent fashion what influences the choice of additional information to include in a given correction response and (b) to enable a system to produce such responses automatically. The former involves identifying possible misconceptions that may lead to such an error, while the latter involves establishing and maintaining a model of things the user correctly knows so as to eliminate them from consideration. One example must suffice (for additional discussion, see McCoy, 1983):

> S: Do you have any liquid assets?
> U: I have a $5K money market certificate.
> S: A money market certificate isn't a liquid asset. Your money is tied up for a period of several years in a money market certificate. Perhaps you meant a money market fund.

If the user's response is understood as an affirmative answer to the system's question, it must be the case that s/he believes s/he has something that can or might be classified as a "liquid asset." However, a money market certificate isn't one. Thus the user's inappropriate response might follow from either:

1. his/her lack of knowledge as to what a liquid asset is (i.e., his/her response might just be a guess, naming some investment s/he knows s/he has);
2. his/her incorrect belief that a money market certificate is a liquid asset (i.e., s/he believes that it has the attributes of a liquid asset);
3. his/her incorrect belief that what she has is a money market certificate (i.e., s/he is confusing what s/he has with a money market certificate).

If the system has no reason to rule out any of these possibilities, it might try to respond to all of them. Thus it addresses point 2 explicitly in the first sentence of its reply and point 1 implicitly in the second (i.e., by contrast, a liquid asset is one that is not tied up for a long period). In its third and final sentence, the system addresses point 3, thereby hitting all bases. Remember that it is still the system's goal to determine if the user has any liquid assets. If his/her money is in a money market fund, then the answer is "yes". This is the type of behavior that McCoy is attempting to support.

3.4. "*Event-related*" Misconceptions

The fourth type of misconception involves events and states and their dependencies over time. It is possible for a user to be mistaken about what can be true now or what could have been true (or happened) in the past, because (a) s/he is unaware of the occurrence of some event or (b) s/he does not know that an event has particular consequences or (c) s/he believes some event has occurred when it hasn't. Again, given a question revealing such a misconception, a simple "no" or "none" answer would just perpetuate it. For example,

> U: Is John registered for CSE220?
> S1: No.
> S2: No. He can't be registered for it because he has already advance placed it.

For the user to have asked this question, s/he must be ignorant of either the event that precluded registering or its consequences. While the system may not be able to determine which, it should provide the user with enough information that s/he can square away the misconception him/herself. (In response S2 above, the general rule "Advance placing a course precludes registering for it", which the user may in fact be ignorant of, is not stated explicitly but is assumed to be derivable from the response.)

The knowledge needed to recognize and square away such misconceptions consists of a knowledge of past events (or states of the database) — often preserved but not accessible to the database system — and of the relationship between past events and what can be true afterwards, including possibly the present. The latter is very much like update constraints used to maintain detabase consistency. However in general, update constraints are not expressed in a form that admits reasoning about possible change. Something more is needed. What we have chosen to use in our own research is an extension of the propositional branching time temporal logic (BenAri, et al., 1981), as documented in Mays (1982, 1983).

Our original impetus into this area was a desire to give a database system the ability to take the initiative and offer to monitor for information of which it was currently unaware. For example,

> U: Has John checked in yet?
> S: No — shall I let you know when he has?

> U: Has John checked in yet?
> S: Yes — shall I let you know when the rest of the committee members do?

Work on producing monitor offers that are both competent (i.e., that correspond to a possible future state of the database) and relevant (i.e., that the user would be interested in) is proceeding concurrently with the work reported on here (Mays, 1982, 1983). We have termed systems which can reason about possible future states of the database "dynamic database systems".

We do not have the space here to explain in detail the logical system we are using (but see Mays, 1982). Briefly, there is a reserved time point NOW, where the past is viewed as a linear sequence of time points (prior to and including NOW) and the future is viewed as a branching structure of time points (following and including NOW). A set of nine complex operators are available to quantify propositions as to the points they are asserted to hold over — e.g.,

$\forall$ Gq–proposition q holds at every point of every future
$\exists$ Gq–proposition q holds at every point of some future
$\exists$ Xq–proposition q holds at the next point in some future
Pq–proposition q holds at some time in the past
etc.

Axioms assert the relationship between events/states in the past, present and future. For example, if the propositional letter 'a' is taken to stand for 'student advance places course' and 'r', for 'student is registered for course', the following axiom states the continuing rule that a student who has advance placed a course (some time in the past) is not now registered for it —

$$\forall G(Pa \rightarrow \sim r)$$

For the above registration example, to distinguish whether it is **accidental** that John is not registered for CSE220 now (he could be, only he's not) or **foreordained** (some event has taken place that precludes registering or some enabling event has not yet occurred) requires the system to suppress its knowledge (or assumption) about John's current status[1] and consider whether it could provably believe the opposite — i.e., that John is registered now for CSE220. If it couldn't, then not only is John **not** registered for CSE220, the system should have identified at least one basis for why he couldn't be. By the above axiom, it is clear that it is not accidental that John isn't registered, because r (being registered for CSE220) is inconsistent with Pa (having advance placed CSE220), that is, $Pa \rightarrow \sim r$.

In summary, we consider it very important for the development of the

[1] which may involve suppressing other facts that follow from his being registered — e.g., that he is taking four courses, that he is in Room 225 from 10–11am MWF, etc. A data dependency scheme such as that discussed in Doyle (1979) or McDermott & Brooks (1982) would be needed to do so.

next generation of cooperative interactive problem-solving systems to enable them to recognize and respond to user misconceptions. In addition to the work discussed above, we have started lines of inquiry into detecting and correcting other types of misconceptions — in particular,

- user misconceptions about how to carry out some procedure (Schuster, 1983), for use in smart Help Systems — for example,

  U: How do I invoke the editor?
  S: You don't have to invoke it. Whenever you're not running some other program you're in the editor.

- user misconceptions about the "best" course of action to achieve some goal (Pollack, et al., 1983).

  U: Do you think it's better to buy T-bonds now or wait a month or two?
  S: In your bracket, you'd be better off buying municipals.

Clearly not all misconceptions can be detected, not all can be easily corrected and, moreover, not all make a difference to the system's ability to convey its information responsibly (Joshi, 1982). What we believe AI can contribute to this area is an improvement on a system's ability to detect and correct misconceptions detrimental to the successful transfer of information from system to user.

4. *Getting Information from Users*

As we noted in the Introduction (section 1), systems must be able to get from their users the information they need to help these users solve their problems. The most commonly used way of getting information is via "menus" — essentially, multiple choice questions. However, there are several problems with relying on menus:

- The user may not understand either the question or the menu options.
- The user may be influenced by the options — i.e., s/he assumes one of them must be appropriate to his/her case, so s/he bends the facts to fit the options.
- The user may not be satisfied with any of the options — i.e., none seems appropriate to his/her case.
- The user may want to qualify his/her response — i.e., s/he may feel that simply agreeing to a particular option will be misconstrued.

In all these cases, the reliability of the user's response is called into question.

For these reasons, we are attempting to provide users with as much freedom in **responding** to questions, ideally in Natural Language, as other systems provide users in **asking** them. This involves at least the following:

1. allowing users more leeway in how they provide the requested information, along with any additional information they believe relevant and want to convey as well.
2. providing the user with more help when s/he doesn't know how to respond.

The first point is important because, ordinarily, people vary in how they respond to questions. Some are direct, some are roundabout, some say a lot, some say a little, some do not know what to say. If systems cannot accommodate this diversity, they will drive users away. On the other hand, there are general patterns within this diversity. For example, one general response strategy involves not only answering the given question but, along with it, providing additional information believed to be relevant to the current (shared) task — e.g.,

> S: Would you like to sit in smoking or nonsmoking?
> U: Non-smoking, on an aisle, near the front please.

As van Katwijk, et al., (1979) has shown experimentally, people are very annoyed if this additional information they offer is ignored. While it is already the case that several frame-based systems (including Bobrow, et al., 1977; Engleman et al., 1980; and Shwartz, 1984) accept volunteered information, there are other significant question-response patterns that a system should also be prepared for.

In the first part of this section, we discuss various common ways in which people respond to questions and the relationship between (a) their **response,** (b) the **answer** to the question and (c) the **additional information,** if any, being offered. We also describe some research being done here whose aim is to enable systems to accommodate one of these response strategies that we see as particularly important. In the second part of this section, we discuss various ways of helping a user who doesn't know how to respond to a question.

4.1. *Responses from which an Answer can be Inferred*

Often a person will respond indirectly to a question, in the belief that the questioner can infer the answer from the response. (We believe the same will be true of users responding to system questions.) We have identified four situations in which this occurs.

1. The user is **unable** to determine an answer to the question but has what s/he believes to be information from which the system can deduce an answer — e.g.,

   S: What is your employee classification: A–1, A–2 or A–3?
   U: I'm an assistant professor in Oriental Studies.
   S: All faculty members are A–1 employees, thank you.

   S: Are you a senior citizen?
   U: I'm 62 years old.

   Of course the user can be wrong, and the information s/he offers either inadequate or irrelevant — e.g.,

   S: What is your employee classification: A–1, A–2, A–3?
   U: I've been here for over 5 years.
   S: Sorry, could you tell me either your job title or position?

2. The user is **able but unwilling** to perform the computations necessary for answering the question. Instead s/he provides data that s/he believes the system can use to compute an answer — e.g.,

   S: What is your yearly salary?
   U: $1840 per month

3. The user decides to be more informative than the question calls for, responding with an instantiation of a general description contained in the question — e.g.,

   S: Did you see any fish?
   U: I saw some guppies.

4. The user feels that a direct answer to the given question would be **logically correct but misleading.** That is, it would imply that there is nothing more s/he can say, relevant to the situation prompting the question. Instead s/he provides **related** information, whose relation to the question s/he believes the questioner can and will recognize — e.g.,

   S: Did you invite the Bennets?
   U: I invited Elizabeth.

   S: Did you invite Elizabeth?
   U: I invited all the Bennets.

   Neither reply answers S's simple yes/no question directly. However, a bare "no" in the first case and a bare "yes" in the second, while literally true, are likely to mislead S as to what is actually true.[2] The given response is more cooperative and moreover more efficient than an explicit statement of both the answer and the added qualification.

The fourth question-response pattern is particularly important for a system to accommodate because we see a user's *fear of being misunderstood* by a computer as being of greater consequence for the future of interactive problem-solving than a user's *annoyance at volunteered information being ignored.* For this reason, it is a current topic of research (Hirschberg, 1983).

As we noted above, the interpretation of responses which fit this pattern requires that the questioner recognize the relation between the particular entity, event or situation referenced in the query and that referenced in the response. From this relation, the questioner can determine not only the respondent's answer to the given question but also the additional information the respondent has included. Two forms of reasoning are involved in determining these two things: standard logical deduction and an extended form of conversational implicature[3] recognized first by Horn (1978) and termed by Gazdar (1975) "scalar quantity implicature".

Consider a speaker whose utterance can be interpreted in terms of an open proposition **P** holding of some scalar value **x.** Horn observed that, following Grice's **Maxim of Quantity** — make your contribution as informative as is required (for the current purposes of the exchange) — the speaker in effect commits him/herself to **x** being the highest value on its scale that **P** holds of, if s/he is observing Grice's **Maxim of Quality** — Do not say what you believe to be false. Propositions formed from **P** by substituting for **x** values higher on the scale (which Horn limits to those that logically *entail*[4] **x**) are thereby implicitly marked — i.e., **implicated** — by the speaker either as 1) not **known** to be the case or as 2) known **not** to be the case, depending upon the discourse context.

A brief example: the lexical items "some" and "all" can be seen as

---

[2] It seems clear to us that the third and fourth response patterns are related. On the other hand, it seems that they can be distinguished by how the response can be paraphrased. Responses following the third pattern can always be paraphrased "Yes, specifically . . .", as in "Are you 65 or older? Yes. Specifically, I'm 72". Not so for the fourth pattern: its correct paraphrases include "No, but . . ." and "Yes, moreover . . .", as in "Did you invite the Bennets? No, but I invited Elizabeth" or "Did you invite Elizabeth? Yes. Moreover, I invited all the Bennets".

[3] An **implicature** is a conclusion that a person would draw from an utterance over and above its propositional content or anything that proposition logically implies. The concept of implicature comes from Grice (1975).

[4] Horn's definition of logical entailment is two-sided: **P** entails **Q** iff $\mathbf{p} \rightarrow \mathbf{q}$ and $\sim\mathbf{q} \rightarrow \sim\mathbf{p}$.

points on a "quantifier" scale, in which "all" entails "some". Thus in the sequence

A: Did anyone leave early?
B: Some people did.

the use of "some" not only explicitly communicates that some people left early but implicates that not all of them did. If all of them did, the cooperative speaker would have committed him/herself to the higher point on the scale. The implicature associated with an utterance of the form "Some X's Y" is thus that "Not all X's Y".

Research on this fourth question-response pattern is being done here by Hirschberg (1983). To begin with, she has extended scalar quantity implicature to include *hierarchical* ordering relations between entities, events and situations (e.g., member-set, part-whole, subtype-type) as well as linear ordering relations, not only entailment but also temporal orderings — both of events and of states — and spatial orderings. For example, in the first exchange above

S: Did you invite the Bennets?
U: I invited Elizabeth.

the two entities (the Bennets and Elizabeth) stand in a set-member relationship. The answer to the system's question follows by implicature: questioning whether an open proposition holds of the set and responding by specifying it holds particular members of that set implicates that the proposition only holds of those members. Next consider

S: Have you dealt the cards yet?
U: I shuffled them.

Here the two events (dealing cards and shuffling them) stand in a linear temporal relationship (i.e., within the process of playing cards). Again the answer to the system's question follows by implicature: questioning one stage in the process and responding with a prior stage implicates that the process has only been taken as far as that prior stage. The latter stage has not yet been reached.

In the examples so far, the answer to the system's question follows by *implicature* from the user's response. Hirschberg has noted other reasoning strategies involved as well. For example, in contrast to the first example above, the question may be whether some property holds of a set member and the response may assert that it holds of the entire set — e.g.,

S: Did you invite Elizabeth?
U: I invited all the Bennets.

Here, the answer to the given question follows from the response by *deduction,* not by implicature (in particular, by "universal" instantiation: if $\forall x.Px$ then Pa). The additional information is that the other members of the Bennet family besides Elizabeth have been invited as well.

This reasoning can get quite complex, employing both deduction and implicature. For example, compare the following two superficially similar exchanges

S1: Did you clean your room?
U1: I made the bed.

S2: Did you clean your room?
U2: I washed the dishes.

The answer in the first exchange follows from the simple whole-part implicature mentioned above. (That is, bed making is part of bedroom cleaning. Hence the answer to S1 is "No, except for the bed".) However, dish washing is clearly NOT part of bedroom cleaning: the answer to S2 is NOT "No, except for the dishes". On the other hand, dish washing is part of house cleaning in general. If the respondent takes this as the question to respond to, his/her response can be understood by implicature as dish washing being the only part of house cleaning that the respondent did. S/he did not do anything else. Hence it follows by deduction that s/he has not cleaned his/her room.

Hirschberg's current research is focussed on developing techniques both for **recognizing** instances of this question-response pattern, using the system's domain knowledge and a model of the user's and system's mutual knowledge, and for **deriving** both the requested information and any additional relevant information from the given response.

This task is made difficult by three things. First, as the above room-cleaning/dish-washing example shows, the process may involve figuring out what question it is that the respondent has chosen to respond to.

Another difficulty is that the second argument to the comparison may not be explicit in the response, but rather may have to be **deduced** from it — as in

S: Did you see the play?
U: I didn't arrive until the intermission.

This can be understood in terms of a whole-part relationship between seeing the play and seeing the first act. (That is, by denying the proposition "seeing X" holds of one part, it follows by implicature that it does hold of the others.) However, that the respondent has not seen the act prior to intermission must be inferred from his/her response. That is,

s/he did not arrive until intermission ==>
s/he was not present before intermission ==>
s/he could not see the act(s) prior to intermission (since being present is a precondition for seeing it)

The third problem is, computationally, the amount of mutual knowledge this type of question-response pattern can draw upon. For example,

S: Did you run to Broad Street?
U: I got to the river.

Here, the direct answer "no" is conveyed by an implicature that depends upon the participants' mutual knowledge of a linear temporal relationship between running to Broad Street and running to the river — specifically, the knowledge that West Philadelphia runners often make a loop that goes east to some point before coming back. Broad Street is east of the Schuykill River. Thus getting to the river implicates not getting further east—i.e., not getting to Broad. While this is a straightforward instance of an answer following from implicature based on a linear temporal ordering of events, recognizing it as such does demand a great deal of domain-specific knowledge that can be assumed to be mutually known.

Despite these problems, the importance we see in enabling systems to understand this type of response to their questions makes it imperative that some sort of solution (preferably a good one!) is devised.

4.2. *When the User Cannot Answer*

When a system makes an attempt to get information from its user, it is very possible that the user will not be able to answer. (As expert systems begin to be consulted by a broader population, their designers will be able to make fewer and fewer assumptions about the knowledge and capabilities of any individual user. Thus it becomes more and more likely that not every user will be able to answer every question s/he is asked.) A flexible system should be able to respond with appropriate information to help the user out of his/her predicament (preferably in the form of a restatement of the question so that the user knows how to respond). There are many reasons why a user might not be able to respond as intended to a question posed by the system:

- The user **doesn't understand** the question — e.g., it contains unfamiliar terms or concepts. Here the question should be rephrased in a way that conveys the meaning of the unfamiliar terms or concepts.

- The user **isn't sure** that the question means the same to the system as it does to him/her. (That is, it contains familiar terms that the user suspects may be used in an unfamiliar way.) In this case, the user needs some alternative description of the question that clarifies it.
- The user understands the question but **doesn't know how** to go about determining the answer. In this case, the system should be able to suggest one or more procedures for doing so.
- The user understands the question but **doesn't remember** the information needed for an answer. In this case, the system should be able to ask the user for information that provides strongly suggestive clues to the information it needs.
- The user understands the question but **doesn't have at hand** the information needed for determining an answer (e.g., lab results). The system should be able to figure out whether it might be able to perform some preliminary reasoning without the missing information and finish things off when it is provided.
- The user **doesn't know why** the system wants the information and won't divulge it until s/he does. In this case, the question needs to be expanded to include the system's reason for asking it.
- The user **doesn't believe** the requested information is **relevant** to solving the problem and is trying to force the expert to adopt another line of reasoning. In this case, the system should be able to identify and pursue an alternative strategy, if one exists. If not, it should be able to explain to the user why it cannot continue. The same holds if it is the case that the user does not wish to divulge the information.

Each of these situations poses a slightly different problem for a cooperative interactive problem-solving system. As things currently stand, rule-based systems built in the Mycin framework (Davis, 1979) allow a user to request justification for why the system has asked its question (WHY?). The system responds with a description of the rule whose left-hand side it was trying to instantiate at the time of asking the question. In addition, work done by McKeown (1982) can enable a system to respond to user requests for term definitions and could be extended to enable the system to rephrase its question in a way that conveys the meaning of the unfamiliar terms. What follows then is a proposal for the set of options a user should have at his/her disposal when s/he is unable or unwilling to answer the system's question, as well as the system knowledge and capabilities required to provide such options:

1. alternate descriptions of terms — e.g.,

   S: Are there other hereditary illnesses that I haven't asked about?
   U: WHAT_IS? "hereditary illness"
   S: Are there any illnesses that pass from parent to child in your family?

2. alternate conceptual models of what those terms might mean to people coming from different backgrounds or levels of expertise — e.g.,

   S: Have you been coughing long?
   U: CLARIFY "coughing long"
   S: Have you been coughing for over three days?

3. procedures for the user's acquiring the information, annotated by the level of resources, time and sophistication needed to carry them out. (This can also be used for guessing which procedures, if any, the user may have already tried, so as to avoid suggesting them.) — e.g.,

   S: Does eating certain foods cause a problem for you?
   U: HOW_KNOW?
   S: Do you get a rash when you eat them or become short of breath or throw up?

   S: Is it an ASCII file?
   U: HOW_KNOW?
   S: Have you ever successfully listed it?
   OR
   S: Was it created by a FORTRAN program using FORMAT statements?

4. "Script"-like information relating events of interest with events that are perhaps more easily remembered by the user (Kolodner, 1981) — e.g.,

   S: Have you had a tetanus shot in the last two years?
   U: DON'T_RECALL
   S: Were you bitten by a dog or cat or some other animal in the last two years?
   U: Yes.
   S: Were you treated for the bite?
   U: Yes.
   S: Were you given a shot as part of the treatment?
   U: Yes.
   S: Then that was probably a tetanus shot.

5. teleological information concerning the reasons for making the request — i.e., the role of the requested information in the reasoning chain — e.g.,

   S: How tall are you?
   U: WHY?
   S: We would like to know whether your weight is right for your height.

6. the ability to do case analysis reasoning if the user can not answer a question. That is, the system can consider the effect on the outcome under the alternative assumption of each possible value or class of values. It is possible that after some analysis the system will discover that it does not make a difference which value is assumed. In the following example, the system could continue by carrying both possibilities forward (i.e., RhFactor ϵ {positive, negative}).

   S: What is your mother's Rh factor?
   U: DON'T_KNOW
   S: Could you ask her? We'll continue now without it.

It is clear to us that cooperative interactive problem-solving systems should be able to provide these capabilities for a user who either cannot or chooses not to answer its questions. We will continue to work on this problem here, and hope that work will be going on elsewhere as well.

5. Conclusion

In this paper we have illustrated additional capabilities that systems need if they are to move beyond straight factual question-answering to participating with their users in cooperative problem-solving interactions. Two of these we have discussed in more detail: getting systems to recognize and respond to users' misconceptions and enabling them to get from their users the information they need to help the users solve their problems. Without these capabilities, interactive problem-solving systems will never be more than laboratory toys.

6. References

Allen, J. Recognizing intentions from natural language utterances. In M. Brody (Ed.), *Computational models of discourse.* Cambridge, MA: MIT Press, 1982.

Ben Ari, M., Manna, Z., & Pneuli, A. The temporal logic of branching time. *Eighth Annual ACM Symposium on Principles of Programming Languages,* 1981.

Bobrow, D., Kaplan, R., Norman, D., Thompson, H. & Winograd, T. GUS, a frame driven dialog system. *Artificial Intelligence 8,* 1977.

Bobrow, R. J. *The RUS system* (Tech. Rep. 3878). Cambridge, MA: Bolt Beranek and Newman, Inc., 1978.

Cohen, P., Perrault, C. R. & Allen, J. *Beyond question-answering* (Tech. Rep. 4644). Cambridge, MA: Bolt Beranek & Newman Inc., 1981.

Davis, R. Interactive transfer of expertise. *Artificial Intelligence* 1979, *12*(2), 121–157.

Doyle, J. A truth maintenance system. *Artificial Intelligence* 1979, *12,* 231–272.

Engleman, C., Scarl, E. & Berg, C. Interactive frame instantiation. *Proceedings of the First National Conference on Artificial Intelligence (AAAI),* 1980.

Gazdar, G. *Pragmatics.* New York: Academic Press, 1975.

Grice, H. P. Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and Semantics.* New York: Academic Press, 1975.

Grosz, B. *The representation and use of focus in dialogue understanding* (Tech. Rep. 151). Menlo Park, CA: SRI International, 1977.

Hendrix, G., Sacerdoti, E., Sagalowicz, D. & Slocum, J. Developing a natural language interface to complex data. *ACM Transactions on Database Systems* 1978, *3*(2), 105–147.

Hirschberg, J. *Scalar quantity implicature: A strategy for processing scalar utterances* (Tech. Rep. MS–CIS–83–10). University of Pennsylvania, Computer and Information Science, May 1983.

Horn, L. Lexical incorporation, implicature, and the least effort principle. *Proceedings of the 14th Regional Meeting, Chicago Linguistic Society,* 1978.

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

Kaplan, J. Cooperative responses from a portable natural language database query system. In M. Brady (Ed.), *Computational Models of Discourse.* Cambridge, MA: MIT Press, 1982.

Kolodner, J. L. Organization and retrieval in a conceptual memory for events or CONS54, where are you? *Proceedings of the International Joint Conference on Artificial Intelligence,* 1981, 227–233.

Kwasny, S. & Sondheimer, N. Relaxation techniques for parsing ill-formed input. *American Journal of Computational Linguistics* 1981, *7*(2), 99–108.

Malhotra, A. *Design criteria for a knowledge-based English language system for management* (Tech. Rep. TR–146). Cambridge, MA: Project MAC. MIT Press, 1975.

Malhotra, A., Sheridan, P. *Experimental determination of design requirements for a program explanation system.* (Tech. Rep. RC 5831). Yorktown Heights, NY: IBM Research Center, 1976.

Mays, E. Failures in natural language systems: application to data base query systems. *Proceedings of the 1980 National Conference on Artificial Intelligence (AAAI),* August 1980.

Mays, E. Monitors as responses to questions: Determining competence. *Proceedings of the 1982 National Conference on Artificial Intelligence,* 1982.

Mays, E. A Modal temporal logic for reasoning about change. *Proceedings of the 1983 Association for Computational Linguistic Conference,* 1983.

McCoy, K. Correcting misconceptions: What to say. *CHI'83 Conference Human Factors in Computing Systems,* Cambridge MA, December, 1983.

McDermott, D., & Brooks, R. ARBY: Diagnosis with shallow causal model. *Proceedings of the National Conference on Artificial Intelligence,* (AAAI), 1982, 314–318.

McKeown, K.. *Generating natural language text in response to questions about database structure.* Unpublished doctoral dissertation, University of Pennsylvania, 1982.

Pollack, M. *A framework for providing appropriate advice* (Tech. Rep. CIS–83–28). University of Pennsylvania, Computer & Information Science, October, 1983.

Pollack, M., Hirschberg, J., & Webber, B. User participation in the reasoning processes of expert systems. *Proceedings of the 1982 National Conference on Artificial Intelligence,* *(AAAI),* 1982. (A longer version appears as Tech. Rep. CIS–82-9. University of Pennsylvania, Dept. of Computer and Information Science, July 1982.)

Robinson, J. DIAGRAM: A grammar for dialogues. *CACM,* 1982, *25*(1) 27–47.

Schmolze, J. G., & Brachman, R. J. Proceedings of the 1981 KL-One Workshop (Tech. Rep. 4842). Cambridge, MA: Bolt Beranek and Newman Inc., 1982. Also FLAIR TR-4, Fairchild Lab for AI Research.

Schuster, E. *Explaining and expounding* (Tech. Rep. MS–CIS–82–49). University of Pennsylvania, Computer & Information Science, 1982.

Schuster, E. Custom-made responses: Maintaining and updating the user model (Tech. Rep. MS–83–13). University of Pennsylvania, Computer & Information Science, September 1983.

Schank, R. C., & Slade, S. Advisory systems. In W. Reitman (Ed.), *Artificial intelligence applications for business.* Norwood, NJ: Ablex, 1984.

Shwartz, S. P. Natural language processing in the commercial world. In W. Reitman (Ed.), *Artificial intelligence applications for business.* Norwood, NJ: Ablex, 1984.

Sidner, C. L. Focusing in the comprehension of definite anaphora. In M. Brady (Ed.), *Computational Models of Discourse,* Cambridge, MA: MIT Press, 1982.

Sidner, C. L. & Israel, D. J. Recognizing intended meaning and speakers' plans. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence,* 1981, 203–208.

Swartout, W. Explaining and justifying expert consulting programs. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence,* 1981.

van Hahn, W., Hoeppner, W., Jameson, A., & Wahlster, W. The anatomy of the natural language dialogue system HAM–RPM. In L. Bolc (Ed.), *Natural Language based Computer Systems.* Munich: Hanser/Macmillan, 1980.

van Katwijk, A., van Nes, F., Bunt, H., Muller, H., & Leopold, F. *Naive subjects interacting with a conversing information system.* Eindhoven, Netherlands: IPO Annual Progress Report, 1979.

Wallis, J., & Shortliffe, E. Explanatory power for medical expert systems. *Methods Inf. Med,* 1982, *21*(3), 127–136.

Waltz, D. An English language question answering system for a large relational database. *CACM,* 1978, *21*(7), 526–539.

Webber, B. L. *A formal approach to discourse anaphora.* New York: Garland Press, 1978.

Webber, B. L. Pragmatics and database question answering. *Proceedings of the International Joint Conference on Artificial Intelligence,* 1983.

Wilensky, R. Talking to UNIX in English: An overview of UC. *Proceedings of the 1982 National Conference on Artificial Intelligence, AAAI,* 1982, 103–106.

Woods, W. A. Natural language communication with machines: An ongoing goal. In W. Reitman (Ed.), *Artificial Intelligence applications for business.* Norwood, NJ: Ablex, 1984.