COLING 82, J. Horecký (ed.)
North-Holland Publishing Company

Taking the Initiative in Natural Language Data Base Interactions: Justifying Why

Bonnie Webber and Aravind Joshi
Department of Computer and Information Science
University of Pennsylvania
Philadelphia PA 19104

I. Introduction

In answering a factual database query, one often has the option of providing more than just the answer explicitly requested. As part of our research on Natural Language interactions with databases, we have been looking at three ways in which the system could so "take the initiative" in constructing a response: (1) pointing out incorrect presuppositions reflected in the user's query [4,5]; (2) offering to "monitor" for the requested information or additional relevant information as the system learns of it [6,7]; and (3) providing grounds for the system's response - i.e., "justifying why". The following responses illustrate "presupposition correction", "monitor offers" and "justification", respectively. This paper describes our research on producing justifications. ("U" refers to the user, "S" to the system.)

U: Is John taking four courses?
S1: No. John can't take any courses: he's not a student.
S2: No, three. Shall I let you know if he registers for a fourth?
S3: No, three - CIS531, CIS679 and Linguistics 650.

Database systems are growing more complex in both their domain models and reasoning capabilities. For many reasons, it is important that such systems be able to point the user to what responses follow from. In this regard, justification differs from argument, whose role is to convince the listener. The role of justification is rather to give the listener an informed basis for either accepting the system's response or questioning its premises or reasoning.

There are at least three issues involved in producing justifications:

(1) what justification does a query call for?
(2) what justification can the system give?
(3) What constitutes a clear and understandable justification?

The first depends primarily on whether or not the user's perceived expectations have been fulfilled by the system's answer. For example, the use of "still" in a query indicates that the user expects that the potentially changing situation described in his/her query has not in fact changed.

U: Does anyone still have an Incomplete in CSE110?
S1: Yes, John does.
S2: No. The last remaining Incomplete was made up on 1 December.

If the user's expectation is correct (i.e., not all Incompletes have been made up), then facts corroborating that expectation may constitute appropriate justification as in the "yes" answer above. If it isn't correct (as in the "no" case), then specifying the event that enabled the change may constitue a more appropriate justification.

More often than not, the user's expectations will not be as clearly identifiable as in the "still" case above. The system will have to formulate and keep a growing model of the user, as its basis for determining what situation the user believes to hold and hence what s/he expects to learn from his or her queries. This is a significant area of research, which we are pursuing in parallel with the work reported on in this paper.

The second issue in justifying a direct answer is what justification the system can give. In the simple case, the system may know no more than the particular facts in its data base. In that case, the only justification the system can provide is those facts themselves (or some description of them). For example, suppose the system knows for each student-course pair only whether the student had passed (P) or failed (F) the course. Then only the following justfication is possible:

U: Did anyone fail CIS531?
S: Yes, John and Martha.

On the other hand, some reasoning may have been involved in deriving the answer, as in the case where the system knows (e.g. has an axiom to the effect) that "failing" a graduate course follows from a receiving a grade of C or worse. In this case, specifying the particular grade each failing student received could be used to justify the system's assertion that they had failed - i.e.,

U: Did anyone fail CIS531?
S: Yes, John received a C and Martha, a D.

Or the system could give some indication of its reasoning as well, as in

U: Did anyone fail CIS531?
S: Yes. With a graduate course, everyone fails who gets a C or below. In CIS531, John received a C and Martha, a D.

We shall claim that possible justifications are related but not equivalent to how the system comes to know an answer. The reason for this reservation has to do with the third issue mentioned earlier in justifying a direct answer and the one we will address further in this paper: that is, what constitutes a clear and understandable justification.

II. Towards Clear and Informative Justifications

The issue of clear and understandable justifications is important everywhere that reasoning is involved in deriving an answer, be it database systems, expert systems or elsewhere. While our theme is extended database question-answering, we believe our points are of general applicability. Where reasoning is involved in deriving an answer, clarity and ease of understanding translate into two goals: (1) succinct justifications that do not say more than is necessary (lest they lose the user) and (2) conceptually well-structured justifications that the user can understand and evaluate. To meet the first goal, we are drawing upon an oft-made observation [2,10,11], first made in Aristotle's Rhetoric, that in exposition, "abbreviated proofs" - what Sadock has called "modus brevis" - are both sufficient and desirable. Our approach to the second goal draws on the strong similarity we see between reasoning and hierarchical planning. Again, while our examples will be drawn primarily from the database domain, our approach, discussed in more detail below, should be of general interest.

A. Succinct Justifications - "Modus Brevis"

As a simple illustration of "modus brevis" and its use in forming succinct justifications, consider a modus ponens deduction, possibly used in deriving an assertion and now to be used in justifying it. It has been observed that in presenting such reasoning one need not make explicit all three parts of the proof - the major premise (A -> B), the minor premise (A), and the conclusion (B). Rather it is sufficient to state the conclusion, with either the major premise or minor premise (but not both) as support. So suppose in response to the query "Did John fail physics?", the system makes the following modus ponens deduction

Anyone who gets below a C fails physics.
(* major premise *)
John got below a C. (* minor premise *)
John failed physics. (* conclusion *)

The system can then justify its "yes" answer in either of the following ways, relying on the user's ability to recognize the underlying deduction.

S: Yes. Everyone failed physics who got below a C.
S: Yes. He got below a C.

"Modus brevis" forms can be used in justifying other types of reasoning as well, both deductive and non-monotonic. However, the speaker must be able to assume that the listener can, on the basis of what is essentially a clue to an argument, reconstruct that argument. On the other hand, whether the listener is convinced by the argument s/he deduces - i.e., whether s/he accepts the inferred premise - is a separate issue: the listener can always initiate a subsequent interaction to confirm that s/he has inferred what the speaker has intended or to question it. For example,

U: Did John fail physics?
S: Yes. He got a B.
U: Is the failing grade really B or below?

Since the successful use of "modus brevis" in justifications depends essentially on the listener's ability to reconstruct an argument from a single clue, it is only used in place of very short reasoning chains. On the other hand, the reasoning we may want to justify may grow quite large and complex. Thus we expect to "modus brevis" forms to reduce the bulk of substructures, rather than for justifying entire arguments. Currently we are cataloguing "modus brevis" forms of various argument types and noting context-dependencies in their use. These schemas will then be used as tools for generating succinct justifications.

B. Clear Justifications - Hierarchical Reasoning

The other goal of our research into producing justifications involves creating text structures which convey appropriate justifications in an understandable way. We have two claims to make here. The first is that just as actions have a hierarchical conceptual organization, so does reasoning - which is essentially the act of supporting or denying a proposition. The former organization can be used in forming plans, revising plans, or describing them to another person [3,9]. Similarly, the hierarchical organization of reasoning can be used both in constructing a proof and in justifying a result. Our second claim is that the computationally efficient reasoning strategies used to prove a proposition (i.e., respond to a query) are not necessarily the best ones to use in justifying a result. What one wants rather is the ability to use the system's reasoning to suggest and instantiate conceptually more accessible strategies for organizing and presenting justifications. Both claims will be discussed in this section.

Many researchers have already observed that explanations have a tree-like structure. This observation reflects a view of each supported assertion as a non-terminal node in a tree, with the sub-tree under it corresponding to the reasons given in its support [2,11]. Since a statement acting as a "reason" may in turn be supported by other statements/reasons, explanations have a recursive structure.

While the above is true, it masks what we see as a more significant recursive organization - one that reflects the inherently recursive strategies that people use in reasoning (i.e., in supporting or denying propositions). These strategies are recursive because they contain subtasks that call in turn for other propositions to be supported or denied. One way to accomplish this is to chose and invoke another strategy. The kinds of strategies we have in mind are things like:

- Simple Backward Chaining - to show that Q is true, find a set of propositions P1,...,Pk from whose simultaneous satisfaction Q follows. For each Pi, show that it follows. Hence Q must be true.

- Simple Case Analysis - to show that Q is true, find some proposition P from which Q follows, independent of P's truth value. Assume P and show that Q follows. Assume ~P and show the same. Since either P or ~P must be true, Q must be true. (Alternatively, to show Q is false, find some P from which ~Q follows, independent of P's truth value. Assume P and show ~Q follows. Do the same for ~P. Since P or ~P, ~Q must be true - hence Q is false.)

- General Case Analysis - to show that Q is true, find some assertion P that is partitionable into P1,...,Pk. Assume each Pi in turn and show that Q follows from Pi. Since some Pi must be true given P is, Q must be true. (This has the obvious complementary strategy for showing Q false.)

- Reduction ad Absurdum - to show that Q is false, find some proposition P whose both assertion and negation follow from Q. Assume Q and show that P follows. Show that ~P follows. Since Q leads to both P and ~P, Q must be false.

(Other strategies are noted in the full version of this paper.) Wherever a strategy calls for showing "P follows" or "~P follows", there another strategy may be chosen and invoked in support. That such strategies are used in reasoning is well-known. What is significant where explanation and justification are concerned is that where the strategy is clear from the text, the explanation or justification is that much easier to follow.

To see this, consider the following tale, whose humor follows in part from the recursive use of simple case analysis in support of successive alternatives.

What is there to be frightened of?

War was on the horizon. Two students in the Yeshiva were discussing the situation.

"I hope I'm not called," said one. "I'm not the type for war. I have the courage of the spirit, but nevertheless I shrink from it."

"But what is there to be frightened about?" asked the other. "Let's analyze it After all, there are two possibilities: either war will break out or it won't. If it doesn't, there's no cause for alarm. If it does, there are two possibilities: either they take you or they don't take you. If they don't, alarm is needless. And even if they do, there are two possibilities: either you're given combat duty, or non-combatant duty. If non-combatant,

> what is there to be worried about? And if combat duty, there are two possibilities: you'll be wounded, or you won't. Now if you're not wounded, you can forget your fears. But even if you are wounded, there are two possibilities: either you're wonded gravely or you're wounded slightly. If you're wounded slightly, your fear is nonsensical, and if you're wounded gravely, there are still two possibilities: either you succumb and die, or you don't succumb and you live. If you don't die, things are fine, and even if you do die, there are two possibilities: either you will be buried in a Jewish cemetery or you won't. Now if you're buried in a Jewish cemetery, what is there to worry about, and even if you are not ... but why be afraid? There may not be any war at all!" [1] p.63

In this example, "there's no call for worry" is the Q meant to be proven. The initial P being used to support Q independent of its truth value is "war will break out". Assuming ~P (i.e., war won't break out), then Q follows because "obviously" ~P -> Q. On the other hand, to show Q follows from assuming P, the speaker invokes a simple case analysis strategy again, this time finding P'- "they take you [into the army]" - meant to support Q independent of its truth value, and so forth.

Our second claim is that the reasoning strategy used to prove some proposition (i.e., respond to some query) is not necessarily the best one to use in justifying the result to the user. What one wants is to be able to use proofs to suggest an appropriate organization of supportable strategies that can be instantiated to form the basis for an understandable justification. Moore's "Blocks World" example [8] provides a good case in point. In this example, there are three blocks A,B and C. A is on B (On A B) and B is on C (On B C). A is green (Green A), C is blue (Blue C) and B's color is not known. It is also the case that whatever is blue is not green and vice versa (ALL x . Green x => ~Blue x). The question is

"Is there a green block on a non-green block?"
(EXIST x,y . Green x AND ~Green y AND On x,y)

Resolution is the only simple machine reasoning method that can find the correct answer "yes" to this problem. (Simple backward-chaining or forward-deduction systems require an additional procedure called "restricted goal/fact resolution".) Converting the above facts and axioms to clause form and using resolution, one proof goes as follows:

(1) ~Green x OR Green y OR ~On x,y	[negation of theorem]
(2) Green A	[axiom]
(3) Green y OR ~On A,y	[resolving 1 and 2]
(4) On A,B	[axiom]
(5) Green B	[resolving 3 and 4]
(6) Green y OR ~On B,y	[resolving 1 and 5]
(7) On B,C	[axiom]
(8) Green C	[resolving 6 and 7]
(9) ~Green z OR ~Blue z	[axiom]
(10) ~Blue C	[resolving 8 and 9]
(11) Blue C	[axiom]
(12) NIL	[resolving 10 and 11]

What this proof does not make obvious is that the answer follows by considering all colors that B can take (that is, green and non-green). Neither does the proof make obvious that the answer follows in either case, even though a different situation holds. That these are the elements of what people give in what they think of as understandable justifications can be seen in protocals that we have collected of people justifying their answers to Moore's problem: most of them do so using a simple case analysis strategy. For example, "Yes - it doesn't matter what color B is. If it's green, then it is the green block on top of a non-green block C. If it's not green, then A is the green block on top of a non-green block B."

Our point is that while resolution theorem provers may be appropriate reasoning engines for data base systems, their proofs do not form a good basis for understandable justifications. Thus at least part of our research is aimed at discovering whether one could recognize in the proof tree of a theorem which of the above understandable reasoning strategies could be in justifying the result and then construct an appropriate valid justification in terms of those strategies.

III. Conclusion

This paper has reported briefly on our research on justification. It is an abbreviated version of our technical report CIS-82-1, which can be obtained by writing to the authors.

The authors would like to thank Barbara Grosz for her helpful comments on previous drafts.

REFERENCES

[1] Ausubel, N. (ed.) A Treasury of Jewish Folklore. New York: Crown Publishers, 1948. (Abridged edition published by Bantam Books, 1980.)
[2] Cohen, R. "Investigation of Processing Strategies for the Structural Analysis of Arguments". Proc. 19th Annual ACL Meeting. Stanford CA, June 1981.
[3] Grosz, B. The Representation and Use of Focus in Dialogue Understanding. Technical Report 151. SRI International, Menlo Park CA, 1977.
[4] Kaplan, S.J. Cooperative Responses from a Portable Natural Language Data Base Query System. Ph.D. thesis. Department of Computer and Information Science, University of Pennsylvania.
[5] Mays, E. "Correcting Misconceptions about Data Base Structure". Proc. 3-CSCSI. Victoria, B.C., May 1980.
[6] Mays, E., Lanka, S., Joshi, A.K. and Webber, B.L. "Natural Language Interaction with Dynamic Knowledge Bases: Monitoring as Response". Proc. 8-IJCAI. Vancouver, B.C., August 1981.
[7] Mays, E., Joshi, A.K. and Webber, B.L. "Taking the Initiative in Natural Language Data Base Interactions: Monitoring as Response". Proc. 1982 European Conference on Artificial Intelligence, Orsay, France, July 1982.
[8] Moore, R. Reasoning from Incomplete Knowledge in a Procedural Deduction System. Technical Report 347. MIT Artificial Intelligence Lab. Cambridge MA, 1975.
[9] Sacerdoti, E. A Structure for Plans and Behavior. New York: Elsevier, 1977
[10] Sadock, J. "Modus Brevis: The truncated argument". Papers from the 13th Chicago Linguistics Society Meeting, Chicago IL, 1977.
[11] Weiner, J. "BLAH, A System which Explains its Reasoning." Artificial Intelligence, 15, 19-48.