Opportunistic Processing in Arguments

Rod McGuire, Lawrence Birnbaum, and Margot Flowers

Yale University
Department of Computer Science
New Haven, Connecticut

In two previous papers, we have proposed a part of a computational theory of argumentation, including representations for argument structure and rules for using those representations in understanding and in rebutting (Birnbaum *et ai* (1980) and Flowers *et al* (1981); related work includes Cohen (1980)). One property of the model which we emphasized is the way in which argument mechanisms and inferential memory can each help to direct the processing of the other. In particular, we presented examples in which inferential memory can uncover good rebuttals to an input as a side-effect of the processing that naturally goes on in trying to understand that input. When such opportunities for rebuttal are noticed during understanding, they render unnecessary the use of argument rules to find a response, since one has already been discovered.

For example, consider the following exchange in a mock argument between an Arab and an Israeli over Middle East affairs:

[I]Arab: Israel is trying to take over the Middle East.

[2] Israeli: If that were our goal, we wouldn't have given back Sinai to the Egyptians.

The Israeli's understanding of the Arab's claim [1] involves instantiating a knowledge structure representing imperialism, with Israel as the actor and the Middle East as the target, and recognizing that this is intended as an accusation. This knowledge structure (let's call it TAKEOVER) has several component substructures, roughly as shown in figure 1.

We propose that in trying to understand input [1]. the Israeli must relate this entire structure to his long-term memory. In so doing, he will discover, among other things.

This work was supported in part by the Defense Advanced Research Projects Agency, monitored by the Office of Naval Research under contract N000I4-75-C-I III, and in part by the National Science Foundation under grant IST79I8463

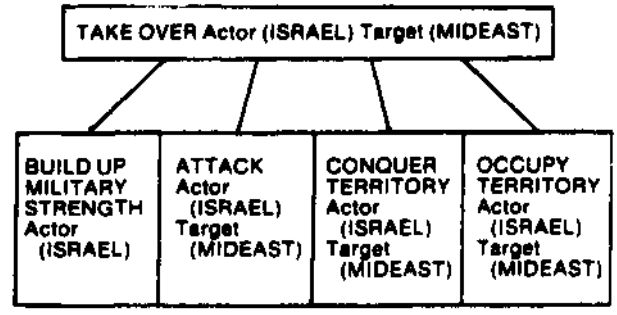

Figure 1

that Israel has indeed engaged in building up its military strength (although in his memory that fact would be explained by the goal of self-defense). More importantly for this example, in the course of relating the OCCUPY TERRITORY substructure to memory, he will find a counterexample - an instance of Israel relinquishing occupied territory (the Sinai).

This fact, which contradicts the original allegation of imperialism, forms the basis of the Israeli's rebuttal [2]. There remains the problem of distinguishing this fact, which is extremely relevant from the point of view of producing a rebuttal, from other facts brought to light while relating the input to memory. Inferential memory must be informed enough about the goals of the arguer to realize that any evidence it uncovers which contradicts the allegation of Israeli imperialism will be useful, and should therefore be saved. This entire process is an instance of the more general phenomenon of reminding (Schank (1980) and (1981)).

As another example of this kind of processing, consider the following continuation of the previous exchange:

(3] Arab: But then why haven't you given back the West Bank to the Palestinians?

Both the Israeli utterance [2] and the Arab response [3] refer to Arab territory occupied by the Israelis. It seems entirely reasonable to suppose that this topic is sufficiently important to an informed supporter of the Arab position to

warrant the existence in his memory of some knowledge structures which organize information relevant to it. In particular, these knowledge structures would point to instances of OCCUPY TERRITORY which have Israel as the actor and former Arab lands as the target. Further, these would be the exact structures which we would logically expect to play a role in the inferential memory processing needed to understand utterance [2]. Thus, in the course of trying to understand the utterance, the Arab would naturally be reminded of instances of continued Israeli occupation of Arab territory. One of these instances (the West Bank) forms the basis of the Arab response [3]. Further examples of this sort can be found in Birnbaum *et al* (1980) and Flowers et *al* (1981).

In many cases, of course, no rebuttal will be uncovered by inferential memory during the understanding phase. It then becomes necessary to utilize argument rules and structures in order to select a point to attack or defend. For example, consider the following argument fragment:

[4] Israeli: The Arabs started the 1967 War, by blockading the Straits of Tiran.

[5] Arab: But Israel attacked first.

[6] Israeli: According to international law, blockades are acts of war.

[7] Arab: Were we supposed to let you import American arms through the Straits?

By our analysis, the Arab's use of inferential memory during the course of understanding the Israeli's claim [6] does not yield a possible rebuttal as a side-effect. Hence, the derivation of his response [7] must result from the explicit application of argument rules based on larger structural features of the entire fragment [4] through [6].

In our model, the structure of an argument is represented by an *argument graph* in which the individual propositions of the argument are related by *support* and *attack* links. For example, the argument graph representation that we propose for the above text fragment is shown in figure 2. Many of the propositions in this graph, for example [4a] and [5a], are not explicit in the utterances given, and must be inferred. The motivation for their presence, and mechanisms for producing them, are discussed in Flowers et al (1981).

The argument graph shown in figure 2 is an instance of a *contrastive positions structure,* an argument form which is generally characterized by a mutual attack relation between two central propositions (in this case [4a] and [5a]) to which further supporting and attacking propositions are attached. Rules associated with argument structures of this sort are used to constrain possible response choices. The rules for a contrastive positions structure suggest two rebuttal options: the Arab may offer additional support for his own claim

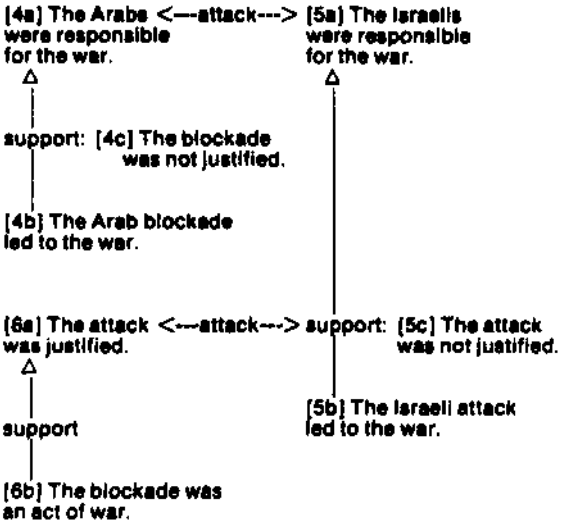

Figure 2

[5a], that the Israelis were responsible for the war, or he can attack the Israeli claim [4a], that the Arabs were. This latter possibility is realized in the Arab's response [7], which attempts to justify the blockade, and thus attacks the support relation between [4a] and [4b]. Although inferential memory is of necessity involved in producing this justification, in this case it plays a secondary role, directed by the argument rules.

These examples illustrate that rebuttals can be produced in two very different ways, either as a side-effect of inferential memory processing performed at understanding time, or as a result of explicit use of argument structures and rules. An important corollary of this processing distinction is that if a direct attack (i.e.. a contradiction) is made on an input, it was discovered at understanding time. The argument is as follows. The same inferential memory apparatus, with the same knowledge base, is used both in understanding and in rebutting. Hence, if inferential memory processing does not uncover any contradictory evidence at the time an input is understood, none will be uncovered a few steps later during response formation at the behest of some argument rule, since exactly the same processing, leading to the same outcome, would occur then. So there is no point in having an argument rule which advises trying to find a direct attack on the input: by the time any such rule were invoked, either the basis for a rebuttal would already be in hand, or no direct attack on the input would be possible.

This point has implications for the role of argument rules in our theory. If direct attacks are only discovered by inferential memory during understanding, then a key function of the argument rules must be to focus attention on

other points of possible contention in the argument when no direct attack on the input is possible. That is. they must primarily be concerned with identifying which previous points are worth going back to, or which new points arc worth raising.

This distinction also has broader implications for computational models of argumentation, and more generally, conversation. Hobbs (1979), among others, has argued that conversation is best viewed as planned behavior, in which utterances are produced by some kind of planning mechanism which is trying to achieve the conversational goals of the participants. Our notion of rebuttals produced as a side-effect of understanding an input implies that any such planning mechanism must be *opportunistic,* in a sense akin to that of Hayes-Roth and Hayes-Roth (1979). That is, it must be able to utilize opportunities for rebuttal which are discovered by inferential memory when performing another task (understanding). It seems possible that a theory of conversation (or more specifically of argumentation) based on this kind of opportunistic processing can reconcile our everyday perceptions of conversations (or arguments) as being, on the one hand, planfu). and on the other, wandering and disorganized.

References

Birnbaum, L., Flowers, M., and McGuire, R. 1980. Towards an AI model of argumentation. In *Proceedings of the First AAA/Conference,* Stanford, CA,pp. 313-315.

Cohen, R. 1980. Understanding arguments. In *Proceedings of the Third CSCS//SCEIO Conference.* Victoria, BC.

Flowers, M., McGuire, R., and Birnbaum, L. 1981. Adversary arguments and the logic of personal attacks. In W. Lehnert and M. Ringle, eds.. *Strategies for Natural Language Processing,* Lawrence Erlbaum Associates, Hillsdale. NJ (forthcoming).

Hayes-Roth, B., and Hayes-Roth, F. 1979. A cognitive model of planning. In *Cognitive Science,* vol. 3, pp. 275-310.

Hobbs, J. 1979. Conversation as planned behavior. In *Proceedings of the Sixth IJCA1,* Tokyo, Japan, pp. 390-396.

Schank, R. 1980. Language and memory. In *Cognitive Science,* vol. 4, pp. 243-284.

Schank. R. 1981. *Dynamic Memory: A Theory of Learning in Computers and People.* (Forthcoming).

NATURAL LANGUAGE INTERACTION WITH DYNAMIC KNOWLEDGE BASES: MONITORING AS RESPONSE*

Eric Mays, Sitaram Lanka, Aravind K. Joshi, and Bonnie L. Webber

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, Pennsylvania 19104

ABSTRACT

In this communication, we discuss an interesting aspect of natural language interaction with dynamically changing knowledge bases - the ability to monitor for relevant future changes in that knowledge. We also indicate the status of our current work in this area and the overall goals of our research on question-answering and monitoring dynamic knowledge bases.

1. INTRODUCTION

The overall goal of this research is support for natural language interactions with a dynamically changing knowledge base.* One significant aspect of this environment is the added dimension along which a system can respond: that is, it can take the initiative and attempt to provide additional information presumed relevant to the user. In particular, the system can offer to inform the user of any information it may learn in the future, that is relevant to the user's query. For example, (S: System, U: User)

U: Did A T & T common stock go up today?

S: Yes. Shall I let you know if it keeps rising?
or
S: No. But shall I let you know if it starts a comeback?

In order to take such initiatives the system must be competent enough to monitor only for conditions which might possibly occur. Of course, this monitoring behavior must also be relevant to the user's intentions. The major issues that must be addressed here are as follows:

1. identifying those properties of actions, events, and processes that influence what monitoring behavior is appropriate, e.g. what aspects of the world may change and hence, what updates to the knowledge base are expected or possible, how those aspects of the world may change and hence, what constraints exist on updates, etc. The latter are also necessary for maintaining the consistency of the knowledge base as updates occur. Given that the event8, processes and actions that the knowledge base reflects may not (or perhaps should not) be modelled as independent, knowing exactly what needs to be monitored will Involve addressing the frame problem as well [7].

2. developing an adequate knowledge representation capable of representing such properties. (Since our primary concern is appropriate monitoring behavior, a full representation of actions, events and processes may not be necessary).

3. developing a language for expressing monitors.

4. identifying natural language cues to the kinds of additional Information a user would find relevant**.

5. designing and implementing a system which exhibits appropriate monitoring behaviors.

We have begun research in these five areas, and the brief examples which follow illustrate not only our current status but also the larger goals of the research on question-answering and monitoring over dynamic knowledge bases.

2. EXAMPLE

For the purpose of illustration, we assume an entity-relationship type of model [3], as illustrated in Figure 1. Entity sets are shown in rectangles, relationships set in diamonds with arrows from participating entity sets, and attributes

Complete knowledge of the user's intentions would clearly help to frame appropriate monitoring behavior. However achieving this unambiguously is a formidable if not impossible problem. On the other hand, it is possible for the system to presume a certain intent and then exhibit the corresponding monitoring behavior. What is necessary here is that the system inform the user of its presumptions so that s/he had the opportunity to correct them.

+ —
This work is partially supported by a grant from the National Science Foundation, MCS 79-08401.

For some work on cooperative responses In static knowledge bases, see [6], [8], and [9]. Some Issues concerning updatability are discussed in [5]. For some work on temporal aspects of knowledge bases, see [l], [2], [4], and [l0].

with dotted links to the entity sets. So, for example, this particular schema shows that some users may run some programs, some have a name and a password, some users register for some courses, etc. The crucial aspect Is that the possible changes in the world represented in this database (i.e., its updatability) can be encoded in the schema. Updatability here corresponds to the ability to update instances of an entity set, relationship set or attribute.

One method of encoding updatability information is by means of the form (square brackets indicate choice and angle brackets, variables).

(UPDATABILITY <entity> [ADD, NO-ADD] [DELETE, NO-DELETE])

(UPDATABILITY <relationship> [ADD, NO-ADD] [DELETE, NO-DELETE])

(UPDATABILITY <attribute> [UPDATE, NO-UPDATE])

These assertions basically state whether or not additions and deletions can be made to a set and whether or not the value of an attribute may be updated. The following updatability assertions might be made for the schema shown in Figure 1.

```
(UPDATABILITY USERS ADD NO-DELETE)
(UPDATABILITY USER-NAME NO-UPDATE)
(UPDATABILITY USER-PASSWORD UPDATE)
(UPDATABILITY PROJECTS NO-ADD NO-DELETE)
(UPDATABILITY PROJECT-NAME NO-UPDATE)
(UPDATABILITY PROGRAMS ADD DELETE)
(UPDATABILITY PROGRAM-NAME NO-UPDATE)
(UPDATABILITY COURSES NO-ADD NO-DELETE)
(UPDATABILITY COURSE-NAME NO-UPDATE)
(UPDATABILITY WORK NO-ADD NO-DELETE)
(UPDATABILITY RUN ADD DELETE)
(UPDATABILITY REGISTER ADD DELETE)
```

However, such assertions are insufficient since updatability may itself change over time. For example, at many universities, a student's ability to change his/her registration status depends on whether it is registration period, break, drop-add period, etc. Thus the updatability of the registration relationship set is better represented

PERIOD	UPDATABILITY
Pre-registration	ADD NO-DELETE
Break	NO-ADD NO-DELETE
Registration	ADD DELETE
Drop	NO-ADD DELETE
After drop	NO-ADD NO-DELETE

Now consider the following question and the various periods during which it may be posed:

U: "Has John registered for CSE110 yet?"

(During pre-registration or registration (add)...)

S: "No, but I'll tell you when he does."

(During break, drop, or after drop (no-add) ..)

S: "No".

Contrast that behavior with

U; "Is John still registered for CSE110?"

(During registration or drop (delete)...)

S: "Yes, but I'll let you know when he isn't."

(During pre-registration, break or after drop (no delete)...)

S: "Yes."

The rule in operation here is to set up a monitor when a situation expected by the user might possibly occur. While the rule is clearly oversimple, it does lead to the relevant behavior. A crucial assumption here is that the length *of* interaction is short enough that updatability will not itself change during the interaction. Although this is adequate for displaying such behaviors during short interactions, say a terminal session of a few hours, it is clearly not the case for those Job functions that span the entire life of the knowledge base. The determination of the interaction length is therefore critical in the inference of appropriate monitors.

In addition to illustrating how outside properties shape appropriate monitoring behavior, the previous examples illustrate two syntactic forms (is/still, has/yet) that suggest the user's interests and hence what monitor might be relevant. That is, "yet" conveys that an event is expected to occur and "still", that a state is expected to continue holding. Appropriate monitoring behavior then in the first case involves an offer to announce the event (or such an event) when it occurs (e.g. "but I'll tell you when he does"), and in the second case, an offer to announce when that state no longer holds (e.g., "I'll let you know when he isn't"). In the absence of such cues as "yet" and "still", the system might query the user about the possibility of setting a monitor. However, to allow maximum flexibility, we include an extra slot in the updatability assertion indicating whether a monitor should be set, not set, or a question posed to the user.

3. SUMMARY

Natural language Interactions with dynamically changing knowledge bases, Invite types of cooperative behavior beyond those needed for interacting with static knowledge bases. We have briefly indicated the status of our work in this area and our larger goals for the research on monitoring aspects of dynamic knowledge bases.

REFERENCES

] Anderson, T.L., The Database Semantics of Time, Ph.D. Dissertation, Computer Science Department, University of Washington, 1981.

] Bruce, B.C., "A model for temporal reference and its realization in a question answering program", Artificial Intelligence, Vol. 3, 1977.

] Chen, P.P.S., "The entity-relationship model — towards a unified view of data", ACM Transactions on Database Systems, Vol. 1, No. 1, 1976.

] Fagan, L.M., VM: Representing Time-Dependent Relations in a Medical SettingT Ph.D. Dissertation, Computer Science Department, Stanford University, Stanford, CA., 1980.

] Joshi, A.K., "Mutual beliefs in question-answer systems", in Mutual Knowledge (ed. N. Smith), Academic Press, to appear in 1982.

] Kaplan, S.J., Cooperative Responses From a Portable Natural Language Data Base Query System, Ph.D. Dissertation, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA.,1979.

[7] McCarthy, J. and Hays, P.J., "Some philosophical problems from the standpoint of artificial intelligence", in Machine Intelligence (ed. B. Meltzer and D. Michie), Vol. A, Edinburgh University Press, Edinburgh, 1969.

[8] McKeown, K.R., "Generating relevant explanations: natural language responses to questions about data base structure", Proc. First Meeting of the American Association for Artificial Intelligence (AAAI), Stanford, California, 1980.

[9] Mays, E., "Failures in natural language systems: application to data base query systems", Proc. First Meeting of the American Association for Artificial Intelligence (AAAI). Stanford, California, 1980.

[10] Sernadas, A., "Temporal aspects of logical procedure definition", Information Systems, Vol. 5, 1980.

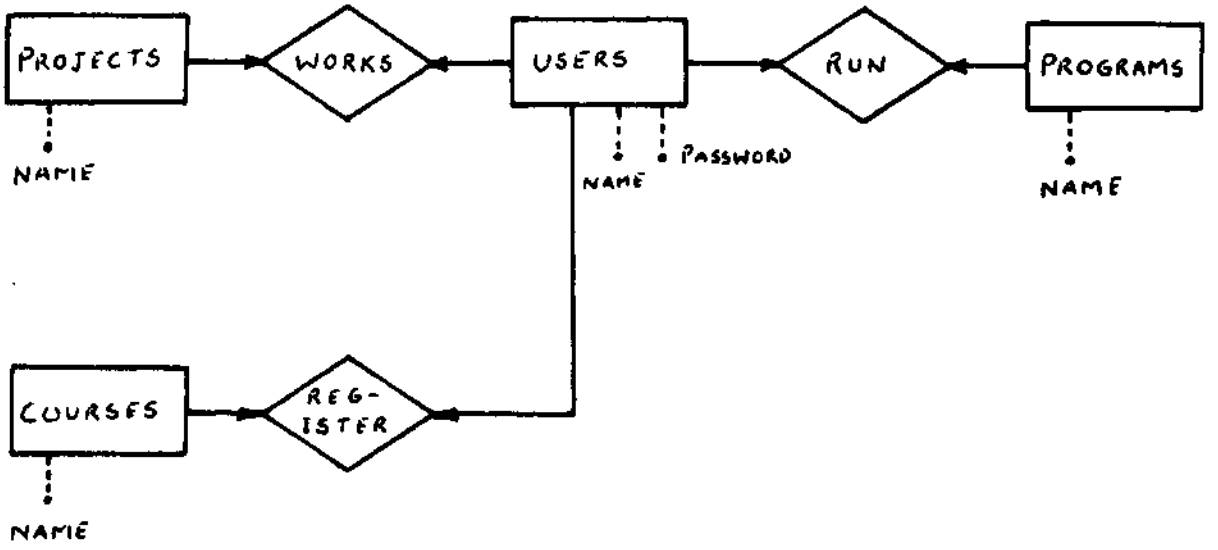

FIGURE 1