TOWARDS AN AI MODEL OF ARGUMENTATIC..

Lawrence Birnbaum, Margot Flowers, and Rod McGuire

Yale University
Department of Computer Science
New Haven, Connecticut

Abstract

This paper describes a process model of human argumentation, and provides examples of its operation as implemented in a computer program. Our main concerns include such issues as the rules and structures underlying argumentation, how these relate to conversational rules, how reasoning is used in arguments, and how arguing and reasoning interact.

Introduction

Engaging in an argument is a task of considerable complexity, involving the coordination of many different abilities and knowledge sources. Some questions that arise in trying to construct a process model of argumentation include:

What sub-tasks comprise argumentation?

What are the rules underlying argumentation?

What representations of argument structure are necessary to support these argument rules?

What are the conversational rules required for dealing with arguments as a specific type of dialogue?

How is the ability to reason about the world used in the argumentation process?

How do reasoning and argumentation interact?

To address these questions, we are in the process of building a system, called ABDUL/ILANA, that can adopt a point of view and engage in a political argument, supporting its beliefs with argument techniques using appropriate facts and reasoning. The program takes natural language input and produces natural language output. All of the rules and mechanisms described in this paper have been implemented, and using them the program is capable of participating in the following argument fragment, concerning the question of who was responsible for the 1967 Arab-Israeli war. The program can assume either the Israeli or the Arab point of view.

[1] Arab: Who started the 1967 War?

[2] Israeli: The Arabs did, by blockading the Straits of Tiran.

[3] Arab: But Israel attacked first.

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

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

[6] Israeli: Israel was not importing arms through the Straits. The reason for the blockade was to keep Israel from importing oil from Iran.

No matter which point of view the program adopts, it uses the same principles of argumentation, abstract knowledge, and historical facts. In this way, the argument is not over the facts but instead over the interpretation of the facts, slanted by the sympathies of each arguer.

Argument tasks

What tasks must an arguer perform when faced with his opponent's responses? Before anything else, each input must be transformed into a meaning representation for use in further processing (see Birnbaum and Selfridge (1980)). But much more is necessary. An arguer must understand how new input relates to the structure of the argument as a whole, and this task usually requires making non-trivial inferences. For example, consider what the program (in the role of an Israeli) must do to relate the Arab's input [3], "But Israel attacked first", to the utterances which preceded it. The Israeli must realize that [3] constitutes evidence for the unstated claim that Israel started the 1967 War, which is contrary to his claim in [2].

An arguer must also relate new input to what he knows about the domain, for two reasons. First, this allows the arguer to determine whether or not

---

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense, monitored by the Office of Naval Research under contract N00014-75-C-1111, and in part by the National Science Foundation under contract IST7918463.

he believes that some claim put to him is true. Second, in doing this he can uncover relevant information which may be useful in forming a response.

Once the understanding sub-tasks sketched out above have been accomplished, an arguer must decide how to respond. This involves several levels of decision. First, he must make large-scale strategic choices, such as whether to attack his opponent's claims, defend his own, change the subject, and so forth. Second, he must determine which points should be taken up, and how evidence can be provided for or against those points. Third, he must use his knowledge and reasoning ability in order to actually produce such evidence. This final step ends with the generation of a response in natural language (see McGuire (1980)).

Argument structures and rules

There are basically two ways that propositions in an argument can relate to each other. One point can be evidence for another point, in which case the argument relationship is one of support, or it can challenge the other point, in which case the relationship is one of attack. Thus, for example, the ultimate analysis of the Israeli's statement:

> [2] Israeli: The Arabs did, by blockading the Straits of Tiran.

requires not only understanding the two separate propositions:

> [2a] The Arabs started the war.

> [2b] The Arab blockade led to the war.

but also realizing that the second proposition [2b] stands in a support relation to the first proposition [2a]. The argument relations are themselves complex structures, including such information as which inference rules enabled the establishment of the relation. (See de Kleer et al. (1977) and Doyle (1979) for a discussion of related techniques.)

One motivation for having these argument relations is that their local "geometry" can be used by argument tactics in determining how to respond to the input. These are rules that describe the options as to how to go about attacking or defending a proposition based on the argument relations in which it takes part. For example, one such rule coordinates the three ways to attack a simple support relationship:

(a) Attack the main point directly;

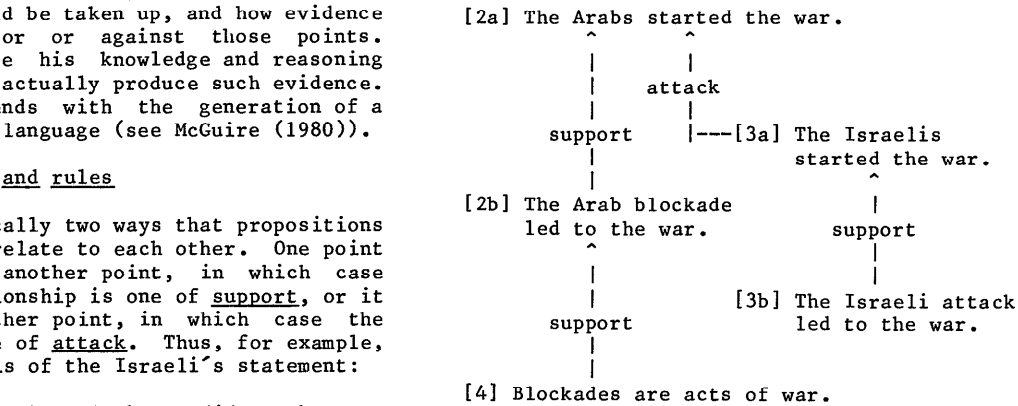
(b) Attack the supporting evidence;

(c) Attack the claim that the evidence gives support for the main point.

The representation of the entire history of the argument, as a network of propositions connected by argument relations, forms an argument graph. Argument graph search rules embody knowledge of how to look through this network for a point which can be attacked or defended. As an example of the use of such rules, consider how the program (operating as the Arab) generates a response to input [4]:

> [4] Israeli: According to international law, blockades are acts of war.

This point stands in a support relation to the previous Israeli claims of utterance [2]:



Once the program has decided to follow an attack strategy, it needs to find a weak point in the opponent's argument. In this case a search rule suggests traversing up the support links in the graph, starting with the most recent input. The first item considered is [4], the Israeli's claim that blockades are acts of war. However this proposition was already checked during the understanding phase and found to be one of the program's beliefs. Thus, it is not a good candidate for attack. Traversing the support link leads to [2b], the proposition that the Arab blockade led to the war. However, this too was verified during understanding. Following one more link leads to [2a], the claim that the Arabs started the war, which the Arab does not believe to be true. Hence, this is a good candidate for attack.

Now the program considers the three tactics for attacking this proposition's simple support relation. Tactic (a), attacking the main point, has already been used once, as can be determined by inspecting the argument graph. Tactic (b), attacking the evidence, can't be used in this case since the evidence has already been rejected as a candidate for attack by the argument graph search rule. This leaves tactic (c), attacking the claim that the evidence is adequate support for the main point.

Having decided to attack this support relation between proposition [2a], that the Arabs started the war, and [2b], that the Arab blockade led to the war, the program now attempts to do that by inferring a justification for the blockade, using its abstract knowledge of blockades and its factual knowledge of U.S. military aid to Israel. This justification is ultimately generated as:

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

An interesting point about question [5] is that it has the form of the standard argument gambit of asking one's opponent to support or justify a position. What makes the question rhetorical is the assumption that, in this case, there is no justification for demanding that arms importation be allowed.

Reasoning and memory in arguments

We have been particularly concerned with investigating how reasoning and memory search interact with the argument process. Reasoning in an argument is not simply blind inference: requirements imposed by the structure of the argument constrain when and how inferences should be made. For example, consider how the program, when adopting the Israeli point of view, responds to question [1]:

[1] Arab: Who started the 1967 War?

[2a] Israeli: The Arabs did,

[2b] Israeli: ... by blockading the Straits of Tiran.

In this case, the generation of [2a] does not require the use of argument rules or episodic memory retrieval. Instead it is derived by use of a "gut reaction" rule that always assigns blame for "bad" events to some other participant. However, as soon as such a claim is put forth in a serious argument, it must be supported. The support [2b] is then produced by a more complex use of inferential memory, activated only in response to the argument goal of providing support.

Conversely, reasoning and memory guide the argument process by discovering information that affects subsequent argument choices. In particular, good rebuttals may often be found in the course of understanding the input. Consider how the program (in the role of the Arab) processes utterance [2]:

[2] Israeli: The Arabs did, by blockading the Straits of Tiran.

To understand this input, the Arab must relate it to what he knows in order to verify its truth, and perhaps more importantly, uncover his relevant knowledge. How does the program verify the claim that the Arab blockade led to the war?

Access to historical events in memory depends upon their organization into temporally ordered chains of related events (see Schank (1979)). These chains are searched by a process called causal-temporal (CT) search. CT search contains the system's knowledge about relations between causality and temporal ordering. Such knowledge includes, for example, the rule that any event up to and including the first event of an episodic sequence can be a cause of that sequence, but no subsequent event can.

The program checks the plausibility of [2] above by employing CT search backwards in time from the start of the war, looking for the Arab blockade. However, in the course of this search it naturally discovers the initial event of the war episode, the Israeli attack, which is then noted as a possible cause of the war. Search continues until the Arab blockade is verified. Then, when the time comes to respond to input [2], the Israeli attack has already been found as a possible rebuttal, so that no explicit search need be initiated.

Thus a prime method for finding possible responses is simply to make use of facts that have been noticed by prior inferential memory processing. This mechanism also enables the system to deny false presuppositions without the necessity for a special rule. This is because in the course of relating input which contains a false presupposition to long-term memory, the program would discover the falsehood -- and often the reason why it's false. Having this in hand, it can then be used in a rebuttal.

Conclusion

Engaging in an argument requires several distinct kinds of knowledge: knowledge of the domain, knowledge of how to reason, and knowledge of how to argue. The proper coordination of these disparate knowledge sources can have a mutually beneficial effect in reducing undirected processing. This kind of result would be impossible in a simplistic model based on strictly separate processing stages.

References

Birnbaum, L., and Selfridge, M. 1980. Conceptual Analysis of Natural Language, in R. Schank and C. Riesbeck, eds., Inside Computer Understanding, Lawrence Erlbaum Associates, Hillsdale, N.J.

de Kleer, J., Doyle, J., Steele, G., and Sussman, G. 1977. AMORD: Explicit Control of Reasoning, in Proc. ACM Symp. on Artificial Intelligence and Programming Languages, Rochester, N.Y.

Doyle, J. 1979. A Truth Maintenance System, Artificial Intelligence, vol. 12, no. 3.

McGuire, R. 1980. Political Primaries and Words of Pain, unpublished ms., Yale University, Dept. of Computer Science, New Haven, CT.

Schank, R. 1979. Reminding and Memory Organization: An Introduction to MOPs, Research Report no. 170, Yale University, Dept. of Computer Science, New Haven, CT.