INVESTIGATION OF PROCESSING STRATEGIES FOR THE STRUCTURAL ANALYSIS OF ARGUMENTS

Robin Cohen
Department of Computer Science
University of Toronto
Toronto, Canada M5S 1A7

ABSTRACT

This paper outlines research on processing strategies being developed for a language understanding system, designed to interpret the structure of arguments. For the system, arguments are viewed as trees, with claims as fathers to their evidence. Then understanding becomes a problem of developing a representative argument tree, by locating each proposition of the argument at its appropriate place. The processing strategies we develop for the hearer are based on expectations that the speaker will use particular coherent transmission strategies and are designed to be fairly efficient (work in linear time). We also comment on the use by the speaker of linguistic clues to indicate structure, illustrating how the hearer can interpret the clues to limit his processing search and thus improve the complexity of the understanding process.

1. BACKGROUND

This paper focuses on one aspect of an argument understanding system currently being designed. An overview of the initial design for the system can be found in [Cohen 80]. In general, we are examining one-sided arguments, where the speaker (S) tries to convince the hearer (H) of a particular point of view. We then concentrate on the analysis problem of determining the overall structure of the argument. Considering an argument as a series of propositions, the structure is indicated by isolating those propositions which serve as CLAIMS and those which serve as EVIDENCE for a particular claim, and by indicating how each piece of evidence supports its associated claim. A proposition E is established as evidence for a proposition C if they fit appropriate slots in one of the system frames representing various logical rules of inference, such that E is a premise to C's conclusion. For example, E will be evidence for C according to modus ponens if E-->C is true. Establishing evidence is a complex process, involving filling in missing premises and recognizing the logical connection between propositions. In any case, our research does focus on reconstructing this logical form of the argument, aside from judgments of credibility.

The initial design [Cohen 80] adopts an unsophisticated processing strategy: each proposition is analyzed, in turn, and each is tested out as possible evidence for every other proposition in the argument. The current design seeks to improve that basic strategy to a selective process where the analysis for a given proposition is performed with respect to the interpretation for the overall argument so far. So, only particular propositions are judged eligible to affect the interpretation of the proposition currently being analyzed. Currently, we assume an "evidence oracle" which, given two propositions, will decide (yes or no) whether one is evidence for the other. With this "accepted" authority, a representation for the argument can be built as the analysis proceeds. (The design of the oracle is another research area altogether, not discussed in this paper).

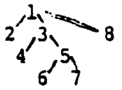
2. THE UNDERSTANDING PROCESS

2.1 PROCESSING STRATEGIES

To process an argument, each proposition is analyzed in turn. It is convenient to think of the representation for an argument as a tree with claims as fathers to their evidence. The speaker thus has a particular tree structure for the argument which he transmits in some order. The hearer must take the incoming stream of propositions and re-construct the logical structure tree. Although the speaker has available a wide variety of possible transmission algorithms, we claim that only a small number of these will be used. We look for transmission algorithms that have associated reception algorithms such that both S and H can process in a reasonable amount of time. Consider the following strategies:

a) PRE-ORDER

The most straightforward transmission for an argument is to present a claim, followed by its evidence, where any particular piece of evidence may, in turn, have evidence for it, following it. A sample tree (numbers indicate order of propositions in the transmitted stream) is:

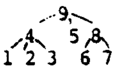


(EX 1)

In this kind of argument, every claim precedes its evidence. Thus, when the hearer tries to find an interpretation for a current proposition, he must only search prior propositions for a father. The reception algorithm we propose for H is as follows: to interpret the current proposition, NEW, consider the proposition immediately prior to it (call it L for last). 1) Try out NEW as evidence for L . 2) If that fails, try NEW as evidence for each of L's ancestors, in turn, up to the root of the tree. (NEW's father must exist somewhere on this "right border" of the tree). When the location for NEW is found, a node for it is added to the tree, at the appropriate place.

b) POST-ORDER

Here, each claim is preceded by its evidence. This is a little more complex for the hearer because he may accept a whole stream of propositions without knowing how they relate to each other until the father for all of them is found. Example:



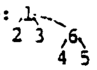
(EX 2)

The reception for H must now make use of the tree for the argument built so far and must keep track of propositions whose interpretation is not yet known, pending the appearance of their father. The formal reception algorithm will thus make use of a stack. Consider L to be the top of the stack. To interpret the current proposition NEW do the following: 1) See if NEW gets evidence from L (i.e. is claim for L). 2a) If L is evidence, keep popping off elements of the stack that are also sons and push the resulting tree onto the stack. 2b) Otherwise, push NEW onto the stack. In short, search for sons; when one son is found, all of them can be picked up. Then the father must stack up to be evidence for some future proposition.

c) HYBRID

Pre-order and post-order are two consistent strategies which the hearer can recognize if he expects the argument to conform to one or the other transmission rules, throughout. But an argument essentially consists of a series of sub-arguments (i.e. a claim plus its evidence). And the speaker may thus decide to transmit some of these sub-arguments in pre-order, and others in post-order, yielding an overall hybrid argument. Therefore, the hearer must develop a more general processing strategy, to recognize hybrid transmission. The reception algorithm now is a combination of techniques from a) and b).

Example:



(EX 3)

But there are additional complications to processing in this model - for example, transitive evidence relations. In EX 3, 4 and 5 are evidence for 1 (since 4 and 5 are evidence for 6 and 6 is evidence for 1), so they will be attached to 1 initially. Then, to process 6, H must attach it to 1 and pick up 4 and 5 as sons. So, the hybrid algorithm involves recovering descendants that may already be linked in the tree.

Here is a more detailed description of the algorithm: We maintain a dummy node at the top of the tree, for which all nodes are evidence. Consider L to be a pointer into the tree, representing the lowest possible node that can receive more evidence (initially set to dummy). For every node NEW on the input stream do the following:

```
forever do
   (B0:) if NEW evidence for L then
    (B1:) if no sons of L are evidence for NEW then
        /*  just test lastson for evidence     */
          (B11:) attach NEW below L
          (B12:) set L to NEW
                 exit forever loop
    (B2:) else
       (B21:) attach all sons of L which are
              evidence for NEW below NEW
        /*  attach lastson; bump ptr. to lastson */
        /*  back 1 and keep testing for evidence */
       (B22:) attach NEW below L
              exit forever loop
   (B3:) else set L to father(L)
  end forever loop
```

This hybrid model still accounts for only some of many possible argument configurations. But we claim that it is a good first approximation to a realistic and efficient processing strategy for arguments is general. It captures the argument structure a hearer may expect from a speaker. Some of the restrictions of this model include: (1) importance of the last proposition before NEW in the analysis of NEW; (2) preference for relations with propositions closer to NEW; (3) considering only the last brother in a set of evidence when NEW seeks to relate to prior propositions. Note then that we do not expect to add evidence for a brother or uncle of L - these nodes are closed off, as only the last brother of any particular level is open for further expansion. To determine the appropriateness of this algorithm as a general strategy, we are currently investigating the implications of restricting expected argument structures to this class and the complexity in comprehension caused by other transmission methods.

Now, the reception algorithms we develop for a), b), and c) can all be shown to work in linear time (the number of evidence relations to be tested will be proportional to the number of nodes in the tree) [see Appendix] but not in real time (can have abritrarily long chains in any sub-argument). Yet hearers process arguments well and this, we claim, is because the speaker helps out, providing special clues to the structure.

2.2 LINGUISTIC CLUES

Special words and phrases are often used by the speaker to suggest the structure of the argument. One main use of clues is to re-direct the hearer to a particular proposition. Phrases like "Let us now return to..." followed by a specific indication of a prior topic are often used in this respect. In EX 1, if 8 is preceded by a clue suggesting its link to 1, then the hearer is spared the long chain of trying 8 as evidence for 7, 5 and 3. So, linear time algorithms can become real time with the aid of clues.

But clues of re-direction may also occur to maintain poorly structured arguments - i.e. the speaker can re-direct the hearer to parts of the argument that were "closed off" in his processing. In certain cases, expectations are then set up to address intermediary propositions. We are developing a detailed theory of how to process subsequent to re-direction.

Another use of clues is to indicate boundaries. In EX 3, if a phrase like "We now consider another set of evidence for (1)..." preceded 4, it would be easier for H to retrieve 4 and 5 as sons to 6 (without checking 3 as well).

Explicit phrases about relations between propositions are only one type of clue. There are, in addition, special words and phrases with a function of connecting a proposition to some preceding statement. These clues aid in the processing of an argument by restricting the possible interpretation of the proposition containing the clue, and hence facilitating the analysis for that proposition. As outlined in section 2.1, the analysis of a proposition involves a constrained search through the list of prior propositions. With these clues, the search is (1) guaranteed to find some prior proposition which relates to the one with the clue (2) restricted even further due to the semantics of the clue as to the desired relation between the prior and current proposition (e.g. MUST be son, etc.). We develop a taxonomy of connectives based on the "logical connectors" listed in [Quirk 72], and assign an interpretation rule to each class.

Notation: in the following discussion S represents the proposition with the connective clue, and P represents the prior proposition which "connects" to S.

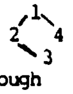
Summary:

CATEGORY	RELATION:P to S	EXAMPLE
parallel	brother	"Secondly"
inference	son	"As a result"
detail	father	"In particular"
summary	multiple sons	"In conclusion"
reformulation	son AND father	"In other words"
contrast	son OR brother	"on the other hand"

Remark: The examples in the following discussion are brief, intended to illustrate the processing issues in argument analysis. We are examining several real life examples from various sources (e.g. rhetoric books, letters to the editor, etc.) but these introduce issues in the operation of the evidence oracle, and so are not shown here.

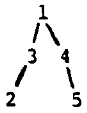
1)Parallel: This category includes the most basic connectors like "in addition" as well as lists of clues (e.g. "First, Secondly, Thirdly,..etc."). P must be a brother to S. Since we only have an oracle which tests if A is SON of B, finding a brother must involve locating the common father first.

EX 4: 1)The city is in serious trouble
2)There are some dangerous fires going
3)Three separate blazes have broken out
4)In addition, a tornado is passing through



The parallel category has additional rules for analysis in cases where lists of clues are present. Then, all propositions with clues from the same list must relate. But we note that it is not always a brother relation between these specific propositions. The relation is, in fact, that the brothers are the propositions which serve as claims in each sub-argument controlled by a list clue.

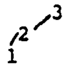
EX 5: 1)The city is awful
2)First, no one cleans the parks
3)So the parks are ugly
4)Then, the roads are ugly, too
5)There's always garbage there



Here, 2 and 4 contain the clues, but 3 and 4 are brothers.

2)Inference: Here, P will be son for S.

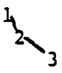
EX 6: 1)The fire destroyed half the city
2)People are homeless
3)As a result, the streets are crowded



Here, the interpretation for 3 only looks to be father to 2.

3)Detail: Here, P will be father to S.

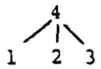
EX 7: 1)Sharks are not likeable creatures
2)They are unfriendly to human beings
3)In particular, they eat people



Here, 3 finds 2 as its father.

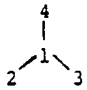
4)Summary: We note that some phrases of summary are used in a reformulation sense and would be analyzed according to that category's rules. These are cases where the summarizing is essentially a repeat of a proposition stated earlier. A "summary" suggests that a set of sons are to be found.

EX 8: 1)The benches are broken
2)The trails are choppy
3)The trees are dying
4)In sum, the park is a mess



But sometimes, the "multiple" sons are not brothers of each other.

EX 9: 1)The town is in danger
2)Gangs have taken over the stores
3)The police are out on strike
4)In sum, we need protection



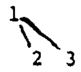
The interpretation rule for summary would follow the general reception algorithm to pick up all sons at the same level.

5)Reformulation: When a clue indicates that S is essentially "equivalent" to some P, P must satisfy the test for both son and father. To represent this relation, we may need an extension to our current tree model (see Section 3 - Future Work).

EX 10: 1)We need money
2)In other words, we are broke

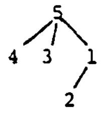
6)Contrast: This category covers a lot of special phrases with different uses in arguments. We have yet to decide how to optimally record contrastive propositions. For now, we'd say that a proposition which offers contrast to some evidence for a claim is (counter) evidence for that claim, and hence S is son of P. And a proposition which contrasts another directly, without evidence being presented is a (counter) claim, and hence S is a brother to P.

EX 11: 1)The city's a disaster
2)The parks are full of uprooted trees
3)But at least the playgrounds are safe



Here, 3 is counter evidence for 1

EX 12: 1)The city is dangerous
2)The parks have muggings
3)But the city is free of pollution
4)And there are great roads
5)So, I think the city's great



Here 3 and 1 are brothers

There are a lot of issues surrounding contrast, some of which we mention briefly here to illustrate. One question is how to determine which proposition is "counter" to the rest of the argument. In EX 12, the proposition with the clue was not the contrastive statement of the argument. So, it is not straightforward to expand our simplified recording of contrast statements to add a "counter" label. Another feature is the expectations set for the future when contrast appears. Sometimes, more evidence is expected, to weigh the argument in favour of one position over another. If these expectations are characterized, future processing may be facilitated.

This description of connective clues is intended to illustrate some of the aids available to the hearer to restrict the interpretation of propositions. We are still working on complete descriptions for the interpretation rules. In addition, we intend each class to be distinct, but we are aware that some English phrases have more than one meaning and may thus be used in more than one of the taxonomy's categories. For these cases, the union of possible restrictions may have to be considered.

2.3 IMPLICATIONS OF THIS ANALYSIS DESIGN

Our description of various processing strategies and clue interpretations can be construed as a particular theory of how to process arguments. The hearer expects the speaker to conform to certain transmission strategies - i.e. does not expect a random stream of propositions. But, H may be confronted with re-directions in the form of special clues, which he interprets as he finds. And he may limit his searching and testing by interpreting clues suggesting either the kind of relation to search for (evidence for, claim for) or the specific propositions to check.

The theory thus proposes a particular selective interpretation process, the techniques are given a formal treatment to illustrate their complexity, and the special markers confronted in analysis are assigned a functional interpretation - to improve the complexity of the understanding task. A note here on the "psychological validity" of our model: we have tried to develop processing strategies for arguments that are consistent with our intuitions on how a hearer would analyze and that function with a realistic complexity. But, we make no claims that this is the way all humans would process.

3. FUTURE CONSIDERATIONS

One area we have not discussed in this paper is that of establishing the evidence relation. For now, the problem is isolated into the "evidence oracle" which performs the necessary semantic processing. In the future, we will give more details on the complexities of this module and its interaction with the general processing strategy described here.

There are, as well, several improvements in processing techniques to consider. Here are some ongoing projects - 1) Investigation of other possible argument structures not included here. The most obvious case to consider is: a claim, both preceded and followed by evidence for it. This is a reasonable transmission to expect. We are working on extensions to the hybrid algorithm to accept these configurations as well. One interesting issue is the necessity for linguistic clues with argument structures of this type - to make sure the hearer can pick up additional evidence and recognize where the next sub-argument begins.

2) Expanding the existing representation model to handle other complications in arguments. In particular, there are several different types of multiple roles for a proposition, which must all be handled by the theory. These include: (i) Proposition is both claim and evidence. (This is already accomodated in our current tree design, where a node can have father and sons). (ii) Proposition is both claim and evidence for the same proposition - i.e. two "equivalent" propositions in the argument. (iii) Proposition is claim to several other propositions. (Again, currently acceptable as father can have any number of sons). (iv) Proposition (E) is evidence for more than one proposition. If all the claims form an ancestral chain - father, grandfather, great-grandfather, etc. then this is just the transitive evidence relation discussed previously and handled by the current strategy. In other cases, (for example, when the claims are brothers) the hearer may not recognize the multiple role in all possible transmissions. For instance, a transmission of claim1, E, then claim2 seems comprehensible. But if the hearer received them in the order: claim1, claim2, then E - would he recover the role of E as evidence for claim1?

3) Trying to characterize the complexity of various argument configurations. Certain combinations of pre and post order seem less taxing to the hearer. We are examining the cases where complexity problems arise and linguistic clues become more prevalent.

4. RELATED WORK

Although our research area may be considered largely unexplored (examining a specific kind of conversation (the argument), concentrating on structure, and developing formal descriptions of processing), there are some relevant references to other work. In [Hobbs 80] Hobbs states that "The problem of AI is how to control inferencing and other search processes, so that the best answer will be found within the resource limitations." We share this committment to designing natural language understanding systems which perform a selective analysis of the input. The actual restrictions on processing differ in various existing systems according to the language tasks and the underlying representation scheme.

In [Grosz 77] focus spaces are used to search for referents to definite noun phrases (and to solve other linguistic problems). These spaces of objects are arranged to form a hierarchy with an associated visibility lattice, based on the underlying structure of the task of the dialogue. Our tree representation is also a hierarchical structure and the description of propositions eligible to relate to the current one may be viewed as a visibility requirement on that hierarchy. So, the restrictions to processing in both our systems can be described similarly, although the details of the design differ to accommodate our different research areas.

In Schank's work on story understanding (e.g. [Schank 75]) stereotyped scripts are used to limit processing. Here, a given proposition is analyzed by trying to fit with expectations for content generated by slots of the script not yet filled. With arguments, we cannot predict future content, so we design expectations that future propositions will have a particular structure with respect to the text so far. These are in fact expectations for coherent transmission. Schank's expectations for coherence, on the other hand, are coincident with his expectations for content, driven by scripts.

Our actual design for restricting analysis is similar in many respects to Hobbs' work on coherence relations ([Hobbs 76], [Hobbs78]). In this work, the representation for the text is also a tree, but the connections between nodes are coherence relations - subordinating relations between father and son, and co-ordinating relations between brothers. In common to both designs is the proposal to construct restricted lists of propositions eligible to relate to a current proposition. In our case, the relations between nodes in the tree is quite different (claim, evidence), although the description for the restricted set turns out to be the same - namely, the right border of the tree.

In Hobbs' system, the search for an interpretation is narrowed by processing a "goal list" of desired relations to existing propositions. We do not have a goal list to order our search, but merely a list of eligible propositions and an ordering of these based on proximity to the current proposition. But we also furnish some motivation for the construction of the eligible list - namely, from the hearer's expectations about transmission strategies used by the speaker.

In addition, Hobbs mentions that a few special words initiate specific goals (for example, "and" suggests temporal succession, parallel or possibly contrast). In our system we also discuss the restrictions to processing furnished by clues but 1) we define the corpus of clues more clearly, indicating several types and their associated restrictions and 2) we make clear the relation between restrictions from clues and the general processing strategy - that analysis picks up clues first, and resorts to general techniques otherwise. Furthermore, we show that a) most classes of clues are simply a restriction on the list of eligible propositions proposed for a general processing strategy and b) certain types of clues may override the general restrictions of the eligible list (e.g. re-directing the hearer explicitly).

ACKNOWLEDGEMENTS

I am gı ıteful to Ray Perrault and Alex Borgida for their suggestions for this paper.

BIBLIOGRAPHY

[Cohen 80]; Cohen, R.; "Understanding Arguments"; Proceedings of CSCSI/SCEIO Conference 1980

[Grosz 77]; Grosz, B.; "The Representation and Use of Focus in Dialogue Understanding"; SRI Technical Note No. 151

[Hobbs 76]; Hobbs, J.; "A Computational Approach to Discourse Analysis"; Dept. Computer Sciences, CUNY Research Report No. 76-2

[Hobbs 78]; Hobbs, J.; "Why is Discourse Coherent?"; SRI International Technical Note No. 176

[Hobbs 80]; Hobbs, J. "Selective Inferencing"; Proceedings of CSCSI/SCEIO Conference 1980

[Quirk 72]; Quirk, R. et al; A Grammar of Contemporary English; Longmans Co.; London

[Schank 75]; Schank, R.; "SAM - A Story Understander"; Yale Research Report No. 43

APPENDIX

Complexity arguments:

PRE and POST ORDER: Any node of the tree is tested to be claim a number of times = #of its sons + 1 more failing test. Now, total tests for claim = "Sum over i" (#sons(i) +1) where i runs over all nodes of the tree, which = "Sum over i"(#sons(i)) + n. But total #sons < total #nodes of tree (no multiple fathers). So total < 2n = O(n).

HYBRID: We measure the complexity of processing all the nodes in the tree, by showing that the #times the algorithm (see section 2.1 for notation) runs through B1, B2 and B3 in total = O(n).

Hypothesis: No node gets attached to another more than twice
Proof: Each NEW gets attached once initially, either at B11 or B22. Once attached, it can only be moved once - in B21, if it is son to current NEW. Once it is moved, it is no longer a son of the current L (since L doesn't change in B2) and can never be son of L again (since L only goes down tree in B12, so never to a previously attached node).

Conclusion: all attachments together are O(n)

Now then, B11 + B22 together are only executed O(n) times - they perform initial attachments. And B12 + B21 must thus also be O(n) - i.e. #times through branches B1, B2 together is O(n).

Now consider B3: here L goes up the tree. But L can only go up as often as it goes down and #moves down tree is O(n) as per B12, so B3 is O(n).

(Note: #tests performed in operations in the forever loop is also O(n) - tests in B0, B1 are just a constant additive factor; #tests in B21 (see comment statement) is $<_2$#attachments in B21).