Interpreting clues in conjunction with processing restrictions in arguments and discourse

Robin Cohen

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Abstract

This paper extends previous work which provided a theory for the interpretation of and necessity for clue words in a particular kind of discourse - namely, one-way arguments. Previous work described a taxonomy of connective clues (words such as "hence" or phrases such as "as a result"), where each clue, classified according to the taxonomy, would set in place a default interpretation of its containing proposition, with respect to the representation for the argument so far. In this paper, we examine how to combine the restrictions for clues with a basic processor for the discourse, offering a integrated processing algorithm, which takes advantage of clues to reduce processing and to detect incoherent arguments, and can still produce an analysis in the absence of clues. We conclude with some suggestions for incorporating clues of re-direction and clues that signal exceptional transmissions. We also demonstrate the implications of our results for discourse in general.

1. Preamble

This paper extends the work of (Cohen 1984) (see also (Cohen 1983)), which provided a theory for the interpretation of and necessity for clue words in arguments. The arguments referred to one-way dialogue where the speaker tries to convince the hearer of a particular point of view.

Previous work described a taxonomy of connective clues (words and phrases), where each clue, classified according to the taxonomy, would set in place a default interpretation of its containing proposition, with respect to the representation for the argument so far. For example, consider the processing of an utterance containing the clue phrase "as a result". "As a result" belongs to an "inference" category, which specifies that the containing proposition must find some prior proposition which supplies evidence to (acts as son to, in the tree diagram for the argument) the containing proposition. (In a sense, this work extended the ideas of (Hobbs 76), where a few special words are shown to signal particular coherence relations in discourse).

The previous paper also discussed the necessity for clue words, describing particular transmissions recognized as exceptional to the basic processing strategy of the argument understanding model, but nonetheless coherent, in the presence of a clue.

This paper first of all addresses the issue of actually processing an argument with clues. We indicate how to combine the restrictions indicated by a connective and its taxonomic interpretation rule with the basic processing strategy, outlined to deal with all arguments (including the cases where no clues exist). We examine tradeoffs in ordering of restrictions suggested by both sources, and propose algorithms for accommodating clue recognition.

We also strengthen our arguments for the necessity of clues with transmissions exceptional to the basic characterization, by illustrating the processing that would occur in the absence of clues. The basic premise is that interpretations with less computational effort would be preferred by a hearer, and would be drawn if clues were not available to override.

The overall conclusion is that clue interpretation processes can be specified, for at least some clues, as a step towards a full processing model of discourse. We are operating in a framework of a model for analyzing discourse by interpreting each new utterance in turn, with respect to the discourse so far. In this sense, each clue provides information for processing, to be integrated into the other tests for interpreting the contained proposition. We will argue for the usefulness of these results for discourse in general.

2. The basic processing algorithm

In order to understand the proposed analysis of clue words in discourse, we offer background on the model for analyzing the structure of arguments, used a basis for our study of clues.

This model (described in more detail in (Cohen 1983), (Cohen 1981)) first proposes that the interpretation for each new utterance in the discourse be done by comparison to a restricted list of prior propositions eligible to relate to a new proposition. The type of discourse is restricted to one turn from a speaker, with a top level goal of convincing the hearer of some point of view (hence, an argument). The representation for the structure of the argument is drawn as a tree, where the relation between a son and its father is one of "evidence". A very simplified summary of "evidence" is that: a proposition P is evidence for a proposition Q if there is a rule of inference such that P is premise to Q's conclusion. The main step in processing is thus to test for possible evidence relations between a new proposition and those already stated, to continue building the tree. The restricted reception algorithm for building the representation is presented below:

```
L: last eligible node; NEW: current proposition
Tree has a dummy root;  succeeds as father to all
(used to simulate a stack)


forever do:
  if NEW evidence for L then
    if no sons of L are evidence for NEW then
   /* just test rightmost son for evidence */
      attach NEW below L
      set L to NEW
      exit forever loop
    else
      attach all sons of L which are evidence
       for NEW below NEW
      attach NEW below L
      exit forever loop
    endif
  else set L to father (L)
  endif
end forever loop
```

This is termed a hybrid reception, because sub-arguments may be inserted claim first (pre-order) or claim last (post-order).

3. Interpreting clues

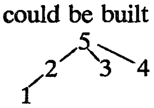
An argument, regardless of presence of clues, is processed in our model according to some proposed restrictions, based on recognizing only coherent transmission orderings from the speaker (as encoded in the algorithm above).

Clue words have been observed to have two functions: to further restrict processing for the speaker, or to signal an exceptional transmission, for the hearer to accept beyond his basic processing strategy. In (Cohen 83) (see also (Cohen 87)) we argue that only certain kinds of exceptional strategies should be accepted as well.

The preferred interpretation will always be one where the basic processing restrictions hold (the hybrid algorithm). To motivate why this is true, consider the example below.

EX1: 1) The park benches are rotting
2) The parks are a mess
3) The highways are run down
4) (Another problem with the parks is that) the grass is dying
5) This city is in sad shape

Without the clue in 4, re-directing to proposition 2, to add more evidence out of turn, a coherent representation could be built just the same, as below:



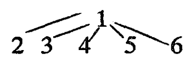
If the speaker intends 4 to add detail to 2, he cannot expect the hearer to recover this structure without a clue, simply because a more effortless interpretation can be recovered as above.

In the absence of a clue word, a coherent interpretation still results, and will be drawn by the model, attempting to satisfy the hybrid constraints. The clue in 4 may signal a different transmission, acceptable because no eligible candidates will otherwise satisfy the semantic constraints of the clue. (Note that 2 is not eligible to receive new evidence, since the hybrid algorithm closes off earlier brothers at a level).

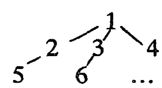
Note as well that with one of the "acceptable exceptional strategies", absence of clues merely produces a different interpretation to the hearer than the one possibly intended by the speaker. In the case of a parallel structure, for example:

EX2: 1) The city has problems
2) The parks are a mess
3) The highways are a mess
4) The buildings are a mess
5) (As for the parks) the benches are broken
6) (And for the highways) there are potholes in the autoroutes...

with a representation, in the absence of clues:



The parallel structure, described in (Cohen 84) as an exceptional strategy, involves a return to a previously closed proposition to add evidence, to then add evidence for each of the brothers of that closed proposition, in turn. The intended representation of EX2, recognizable with clues is:



Since following the hybrid algorithm is basically the preferred interpretation, it makes sense that the restrictions embodied by this algorithm govern the processing of clues. We will first study particular classes of the taxonomy of connectives and propose "processing mesh with the hybrid" for each case. We can then reflect on what the relationship between clue processing and basic search is. We will discuss clues of re-direction (as in EX1) above briefly after the study of connective clues.

4. Integrating connective clues

When clues appear in an argument, these should signal to the basic processor that additional information is being provided by the speaker. This information further restricts the tests for determining the interpretation of the proposition containing the clue. Connective clues provide the additional information of HOW the proposition relates to some prior proposition (see the default rules provided with the taxonomy; the categories are drawn from (Quirk 72)). In the table below, S represents the proposition with the clue; P is the prior proposition which "connects" to S.

Part of taxonomy of clue words, from (Cohen 83)

category	relation: S to P	example
parallel	brother	in addition
detail	son	in particular
inference	father	as a result
summary	father to multiple sons	in sum

We envision a general system architecture consisting of (i) a proposition analyzer, which performs the basic processing algorithm (ii) a clue interpreter, which is called when a clue is detected, and then controls the proposition analyzer (iii) an evidence oracle, which is passed two propositions by the proposition analyzer and responds yes or no whether one is evidence for the other. Since the oracle has a difficult task, the overall efficiency of processing would be improved if either calls to the oracle were avoided, or additional information were available to the oracle to facilitate its testing. (Note that the "oracle" is eventually given some specifications, and is more than just a black box. The processing of the oracle is another topic altogether; see (Cohen 83) for more details).

Our research on integrating clue interpretation with the basic processor is still in progress, but we offer the following algorithm as a first version. Note that this algorithm would then replace the basic processing algorithm (described in section 2). We will explain the main features of the algorithm after its listing.

clue1: true if proposition has a "parallel" clue;
clue2: for detail, clue3: for inference, clue4: for summary

```
forever do:
  /* before testing NEW for L */
  if L=dummy then
    if clue2 then  ((1))
      INTERRUPT-DISCOURSE (and exit loop)
    endif
  endif
  /* see if rightmost son exists */
  if (clue1 v clue3 v clue4) & no
   rightmost son of L then
    if L=dummy then
      INTERRUPT-DISCOURSE ((2B))
      (and exit loop)
    else
      set L to father of L ((3))
    endif
  endif
  if NEW evidence for L then
   /* see if sons will re-attach */
    if no sons of L evidence for NEW then
      if (clue3 v clue4) then
        if L=dummy then
          INTERRUPT-DISCOURSE ((2))
          (and exit loop)
        else
          set L to father of L
        endif
      else
        /* normal attaching */
        attach NEW below L
        set L to NEW
        exit forever loop
      endif
    else  /* some son wants to re-attach */
      attach all sons of L which are
       evidence for NEW below NEW
      attach NEW below L
      exit forever loop
    endif
  else /* if NEW not evidence for L */
    set L to father of L
  endif
end forever loop
```

The first point is that some calls to the evidence oracle can be avoided, if one follows the restrictions of the clue interpretation rules for the taxonomy. Consider the following example:

EX3: 1) The city is in serious trouble
2) There are some fires going
3) Three separate blazes have broken out
4) In addition, a tornado is passing through

The clue in 4 requires 4 to be a brother of some prior proposition. This is realized in our processing model by finding a father from which an attached son may serve as brother.

The hybrid algorithm would have 4 first test to be son to 3 (the last eligible). Since a simple test can confirm that 3 has no sons, it is not considered at all. Thus, one possible call has been avoided, due to the presence of the clue. (This is illustrated in part ((3)) of the algorithm above).

We now consider incoherent arguments. We can specify criteria for recognizing an incoherent transmission, which would then be detected earlier than if the clue did not exist to constrain the required relationship. For instance, in the case where we expect a son prior in the argument, if all tests for father that can also pick up a son fail (must now be at the dummy top to realize this) then we can label the argument incoherent and interrupt - the expectation of the clue has not been met.

Without a clue, we could expect to find later propositions acting as son to current; as such, we would not detect incoherence until the end when no common father exists at the top.

EX4: 1) The parks are a mess
2) The park benches are a mess
3) The playgrounds are a mess
4) The highways are a mess
5) The buildings are a mess
6) The stadiums are a mess

If at this point the argument ended, the analyzer could detect a lack of top level father to detect incoherence.

EX5: 1) The parks are a mess
2) The park benches are a mess
3) The playgrounds are a mess
4) As a result, the highways are a mess

Here, 4 requires a son prior in the discourse. As this fails, the incoherence of this possibly continuing argument (as in EX4) would be detected earlier.

This is incorporated into the algorithm in the tests labelled ((2)). Both parallel, inference and summary require a son (either to attach to NEW or serve as a brother to NEW). If this test cannot be met, the argument is incoherent (see ((2B))). Likewise, if no prior proposition exists to connect to the proposition with a connective clue, regardless of the relationship expected, the argument is again incoherent and the hearer would interrupt (as in part ((1)), where a detail clue expects a non-dummy father prior in the argument).

Examining when an argument is incoherent is also important for studying when clues are used to signal exceptions, rather than just to additionally constrain the basic hybrid case. So, the clarification of when connectives fail in their default interpretations is important as a processing indication to then test for exceptional strategies. (The semantics of the clue and the representation of propositions is also critical; see discussion in section 5).

Are there additional constraints to processing that clues can provide? One possibility we examined was whether some connective clues suggest altering the order of tests performed by the hybrid algorithm. We decided that the order of nodes visited from the eligible list should not change (connectives merely indicate HOW, not WHERE propositions relate). But we examined the effects of testing for a son before testing to be a son at any given node in the tree.

To explain, the inference class, for example, requires a son to be found earlier in the tree. As each eligible node L is examined, should we test sons of L as son to NEW before we test NEW son to L?

Our conclusion is that it is costlier to test for sons first. Defense of this conclusion is offered below.

The standard algorithm, when we are dealing with a statement with an inference clue, can be stated as follows:

```
do
  if L is_father_of NEW then
    attach NEW as son of L
    re-attach sons of L below NEW
    BREAK
  else
    set L to father_of(L)
  endif
enddo
```

If we modify this to check first for a son of NEW, then we have:

```
do
  if NEW is_father_of rightmost_son(L) then
    if L is_father_of NEW then
      attach NEW as son of L
      re-attach sons of L below NEW
      BREAK
    else
      set L to father_of(L)
    endif
  else
    set L to father_of(L)
  endif
enddo
```

Suppose Li is the father to NEW. Under the standard method the following tests will be performed:

```
NEW is_evidence_for L1   -> FAIL
NEW is_evidence_for L2   -> FAIL
...                         ...
NEW is_evidence_for Li-1 -> FAIL
NEW is_evidence_for Li   -> SUCCEED
(then re-attach sons of Li)
```

If we use the modified algorithm, and test for a son of NEW first, then we have:

```
L1 is_evidence_for NEW   -> SUCCEED *
NEW is_evidence_for L1   -> FAIL
L2 is_evidence_for NEW   -> SUCCEED *
NEW is_evidence_for L2   -> FAIL
...                         ...
Li-1 is_evidence_for NEW -> SUCCEED
NEW is_evidence_for Li   -> SUCCEED
(then re-attach sons of Li)
```

The above tests marked * all succeed because of the transitive nature of the evidence relationship. That is, since Li-1 is evidence for NEW, anything which is evidence for Li-1 will also be evidence for NEW. Thus, any test for an Lj to be a son of NEW (with $j < i$) will succeed.

From this we can rewrite the modified algorithm. It is essentially:

```
do
  evidence_oracle call which always succeeds
  if L is_father_of NEW then
    attach NEW as son of L
    re-attach sons of L below NEW
    BREAK
  else
    set L to father_of(L)
  endif
enddo
```

Thus, this algorithm will use more evidence oracle calls than the standard method of checking NEW to be a son of L first. In fact, trying to find a son of NEW first will take on the order of twice as many calls. In short, we adhere to the basic algorithm's testing of NEW to be son, before testing to re-attach propositions to be sons of NEW, regardless of clue.

For the taxonomy classes of detail, inference, summary and parallel (conjunction type versus list type (first, secondly, etc.)), we offer the following results: (i) for these classes, it is not effective to alter the tests at a particular node (ii) it is possible to cut one test to the oracle (iii) one additional advantage that the connective clues provide is to detect earlier incoherent arguments from a speaker (if the expectations associated with the clue are not satisfied by some prior proposition as required).

5. Re-direction clues and future work

Clues which re-direct the processing should have the following relationship to the hybrid: (i) can alter the order of nodes visited (ii) unless the clue also has a kind of connective specified, cannot alter the order of testing at a node or add constraints to the node (i.e. must have sons).

Clues such as "first, secondly, etc." can now be examined as a re-direction indication. They are parallel connectives, expecting brothers prior in the argument, but they also expect to connect at a particular location - namely, at the head of the sub-argument tagged by the specific clue that is one earlier in the list (e.g. "thirdly" expects to connect to "secondly"). For future work, this kind of clue word should be examined to lead in to an incorporation of re-direction clues into the processing algorithm.

In general, re-direction clues are supposed to provide some insight into which prior proposition relates to the one with the clue. Connective clue words only indicate which relation holds with some prior proposition. It is worth investigating how the proper prior proposition can be selected, especially in exceptional cases which override the eligibles for the hybrid. This research will require a deeper investigation of the semantic representation for propositions, used in the analysis.

Another consideration for future work is the role that clue words have on the work of the oracle. In particular, if a connective clue carries certain semantic constraints, how are these precisely communicated to the oracle to facilitate its processing? The answer is obviously influenced by the underlying representation used for the knowledge bases accessed by the oracle and the form of the propositions, when "parsed", made available to the oracle.

We are currently developing an implementation of the algorithm described here to incorporate clues, together with upcoming solutions for handling other kinds of clues, building on the initial implementation of the basic processor, completed in (Smedley 86).

Refinement of the clue interpretation rules and the integrated algorithm is another topic for future work. In (Cohen 83) we offer some motivation for why the interpretation rules as formulated hold for the associated class of clues in the taxonomy. In developing an algorithm for implementation, additional constraints and characterizations may occur. We include a brief discussion of two additional constraints to investigate.

With a clue of the "parallel" category, a brother earlier in the discourse must be found. (Note: it is still coherent to have the father not yet appear in the discourse). According to our integrated algorithm, it is possible for the proposition with the clue (NEW) to find a father (L) and to re-attach the sons of L. Some modification to this test must be made to prevent all the sons of L from re-attaching, thus leaving no brother for NEW. However, it is worth studying whether re-attachment of sons of L is in itself a signal of incoherence.

For the "summary" category, more than one son is to be found earlier in the discourse. So, the integrated algorithm should have an additional test to ensure that when sons are re-attached, more than one re-attaching occurs. But what of the case when the son that re-attaches is in effect a tree, so that there are "multiple sons" for NEW, but not all at the same level? One interpretation is that this structure is, as well, incoherent.

The general problem raised by these suggestions for incoherence is how to consider the interaction between different types of clue words, when more than clue word occurs, either within one sentence or between two sentences which are being tested for a relation (e.g. So, for example... or So, next...). The interacting occurrences may allow for certain relations to be tested in the integrated algorithm, which on the surface seem indicators of incoherence. Studying how multiple constraints may be satisfied is again a topic for future research.

6. Summary

We have provided some new insights into how to incorporate clue interpretation into our model for analyzing arguments, to mesh with the basic processing restrictions. In the process, we have discovered some worthwhile properties of clues: (i) they signal overrides to the processing (for exceptional transmissions) (ii) they provide additional information on where to process or which relationship to find in the prior argument (iii) indications of which relation to find do not constrain the basic processor, except to rule out one test at the last proposition (possibly) or in cases where the argument is incoherent.

We feel that these results carry over to the case of discourse analysis in general. If coherence constraints for processing of discourse are postulated, the clues should help constrain further. Other researchers have studied the role of clue words in discourse (e.g. (Reichman 81), (Grosz and Sidner 85), (Polyani and Scha 83)). If one allows a processing of discourse that does not contain clues, one must comment on how the presence of clues alters the basic processing. In this paper, we suggest how a clue interpretation module would constrain the processor for certain connectives, and point to ongoing work on the analysis of re-direction clues. We also provide insight into when an argument is considered incoherent, and when exceptional transmissions are recognizable (when the clue exists by necessity).

But most of the saving in processing for connectives should come from demanding more specialized semantic relationships (the part tested in our model by the oracle). We have to describe more precisely these operations for future work, to also gain insight into interpreting re-direction clues. We feel that current studies of intonation as a clue (Hirschberg and Pierrehumbert 86) can be treated in a similar fashion. We would then propose an analysis in terms of operations saved, on average, when clues indicate where to test for relations.

Acknowledgements

I am indebted to Trevor Smedley for discussions on this research and comments on earlier drafts of this paper. This research was supported by Nserc (Natural Sciences and Engineering Research Council of Canada).

References

(Cohen 81) Cohen, R.; "Investigation of Processing Strategies for the Structural Analysis of Discourse"; Proceedings of ACL81, 1981.

(Cohen 83) Cohen, R.; "A Computational Model for the Analysis of Arguments"; University of Toronto Computer Systems Research Group Technical Report No. CSRG-151, 1983. (Ph.D. thesis)

(Cohen 84) Cohen, R.; "A Computational Theory of the Function of Clue Words in Discourse"; Proceedings of COLING84, 1984.

(Cohen 87) Cohen, R.; "Analyzing the Structure of Argumentative Discourse"; to appear in Computational Linguistics, 1987.

(Grosz and Sidner 85) Grosz, B. and Sidner, C.; "The Structures of Discourse Structure"; Bolt, Beranek and Newman (BBN) Report No. 6097, 1985. (also Report No. CSLI-85-39)

(Hirschberg and Pierrehumbert 86) Hirschberg, J. and Pierrehumbert, J.; "The Intonational Structure of Discourse"; Proceedings of ACL86, 1986.

(Hobbs 76) Hobbs, J.; "A Computational Approach to Discourse Analysis"; City University of New York Department of Computer Sciences Research Report No. 76-2, 1976.

(Hobbs 78) Hobbs, J.; "Why is Discourse Coherent?"; SRI Technical Note No. 176, 1978.

(Polyani and Scha 83) Polyani, L. and Scha, R.; "On the Recursive Structure of Discourse"; in Connectedness in Sentence, Discourse and Text, K.Ehlich and H. van Riemsdijk, eds., 1983.

(Quirk 72) Quirk, R. et. al.; A Grammar of Contemporary English; Longmans, 1972.

(Reichman 81) Reichman, R.; "Plain Speaking: A Theory and Grammar of Spontaneous Discourse"; BBN Report NO. 4681, 1981.

(Smedley 86) Smedley, T.; "An Implementation of a Computational Model for the Analysis of Arguments: An Introduction to the First Attempt"; University of Waterloo Department of Computer Science Technical Report No. CS-86-26, 1986.