INTEGRATION, UNIFICATION, RECONSTRUCTION, MODIFICATION

An Eternal Parsing Braid

Michael G. Dyer

Computer Science Department
Yale University
New Haven, CT 06520

Abstract

BORIS is an integrated natural language understanding system for narratives. In an integrated system, processes of event assimilation, inference, and episodic memory search occur on a word-by-word basis as parsing proceeds. "Parsing" here refers to the task of building a conceptual representation for each natural language expression. In addition to being integrated, the BORIS parser is also a unified parser. The same parser is used both at story understanding time and question answering time. This paper explores some of the consequences which arise when the same parser serves both tasks. For instance, one such consequence is that BORIS often knows the answer to a question before it has completely understood the question.

L Background

Early natural language programs at Yale used a single parser which operated in isolation from other conceptual processes. I he parser would first map each sentence into its Conceptual Dependency (CD) representation [13], which a script applier (SAM [2]), plan applier (PAM [15]), or question answerer (QUALM [10]) would then interpret. For a time, it was hoped that a single parser could be constructed to serve as a "front-end" for every new project taking natural language as input.

But eventually it became clear that this type of modularity had its drawbacks. With an isolated parser, other process knowledge arising from the parsing task could never be made available as an aid in the natural language understanding task itself. Modularity was abandoned with the creation of FRUMP [3], which relied heavily on scriptal knowledge to direct its parsing processes. Since then every parser at Yale has been integrated to a greater degree - each trying to make use of task and domain specific knowledge to aid in parsing.

To date, BORIS [4] [9], an experimental program designed to understand complicated narratives (approximately 300 words in length), is the most integrated parsing project at Yale. BORIS runs as a single module: all tasks such as event assimilation, inference, memory search and question answering, occur as an integral part of the parsing process. This paper discusses some of the issues and consequences that arise from an integrated approach to text processing.

This work supported in part by Advanced Research Projects Agency contract NOOI4-75-C-111 and National Science Foundation contract IST79I8463.

2. *Integrating Parsing With Memory Search*

Our interest in making BORIS more integrated first arose within the task domain of question answering (QA). An informal analysis of question answering protocols revealed that people were making use not only of general syntactic, semantic and lexical knowledge to parse a question, they were also making use of episodic knowledge -- i.e. the information specifically contained within the story they had read. This made perfectly good sense. If people understand in context, then of course they would make use of the story itself as a context for understanding a question about the story. Here was all of this episodic information available. Why not use it to help parse questions?

For instance, suppose we read the following story:

> Richard hadn't heard from his friend Paul in years. Richard owed Paul money which he had never returned because he didn't know where Paul lived. Then he got a letter from Paul asking him to represent Paul in a divorce case.

Now, if we are asked:

Q: Why hadn't Richard paid Paul back?

we will think of the following answer:

A: Richard didn't know where Paul lived.

Suppose, however, the story had been:

> Richard hadn't heard from his friend Paul in years. Paul had killed Richard's wife and Richard vowed that he would get back at him if it was the last thing he ever did, but Paul had escaped to Argentina.

Now if asked:

Q: Why hadn't Richard paid Paul back?

we do not interpret the question to mean: "Why hadn't Richard returned money to Paul?" Now we understand "pay back" in terms of revenge.

The expression "pay back" cannot be disambiguated by means of general semantic, syntactic or lexical knowledge. The only way "pay back" can be understood in the question above is by searching episodic memory while the question is being parsed. This view of the question understanding process has four major consequences:

(I) Integrated parsing changes our perspective on what it means to understand a question. It no longer means building a representation of the question in isolation of the narrative context. Understanding a question now involves finding a referent for the question in episodic memory. In the terminology of a prior question answering model [10], the parser is now also involved in memory search for an answer key. This means that the representation of a question is no longer just a Conceptual

Dependency [13] diagram but includes pointers into episodic memory - i.e. references to episodic instantiations or higher level knowledge structures.

The process of understanding a question cannot be separated from the process of searching episodic memory.

(2) By viewing parsing and episodic search as an integral process, specific knowledge contained within the narrative becomes available to aid in the parsing process itself. As we have already seen with the example of "pay back", this integrated approach makes 'disambiguation'easier.

Integrating parsing and memory search makes parsing tasks easier.

(3) Since an integrated parser starts searching episodic memory immediately (as soon as the first word of the question is parsed), it is now possible to retrieve the answer to the question before its parse i$ finished. This may sound magical, but in fact people do it all the time. People search memory to find an answer as soon as possible, and then they monitor the rest of the question to make sure that no presuppositions have been violated. For example, given the following story:

> On a night when there was a full moon. John took Mary for a walk along the river. There he kissed her and asked her to marry him. She was so surprised that she was left speechless.

when asked the question:

> Q: Who kissed Mary by the river
> when the moon was full?

most people have the answer "John" by the time they have processed the first three words in the question. The rest of the question is simply monitored to make sure that any presuppositions in the question do not violate knowledge of the story. Earlier Yale systems would have first parsed the question in its entirety before attempting any search for an answer

In an integrated parser, the answer to a question may be known before the question has been completely parsed.

(4) In those cases when an answer is not found before the question has been fully processed, subsequent answer retrieval will be faster in an integrated system. This effect is the result of having searched episodic memory during question understanding. Since the parser has already searched episodic memory, the answer retrieval phase is already partially accomplished by the time the parse phase is completed.

Integrated parsing of questions results in a more efficient answer retrieval.

3. Parser Unification

BORIS did not start out as a single module. Largely for pragmatic reasons, early versions of BORIS had two parsers. One was called DYPAR [5] and was used to parse natural language questions about narratives. The other was a version of CA [I]. The CA parser was used to analyze sentences occurring within narratives and produce a Conceptual Dependency representation for each. At story understanding time, a separate event assimilation module accepted these Conceptual Dependencies as input and constructed an episodic memory of the events in the narrative.

The integrated approach to parsing questions was sufficiently successful in BORIS for us to consider this approach for parsing narrative sentences at story understanding time. An additional motivation rose from our belief that two distinct parsers are not psychologically valid. It seems most unlikely that people would use one parser for answering questions which is fundamentally different from the parser used for understanding stories.

Because earlier parsers had operated in isolation, parser unification had never been an issue. When a parser does not interact with any other processes during sentence analysis, it is trivial to use the same parser to produce Conceptual Dependency representations for both questions and narrative sentences. Parser unification only becomes tricky when parsing procedures are integrated with other memory manipulations.

Recall that, as the CA parser produced a Conceptual Dependency representation as output, the event assimilator then had the task of building episodic memory from each Conceptual Dependency input. Since the parser worked in isolation, it was up to the the event assimilator to apply top-down expectations across sentence boundaries in order to 'knit'events together into a cohesive episodic memory for the narrative. For example, whenever the parser failed to fill a role binding (because of ellisions), or whenever there was a need to interpret A Conceptual Dependency structure in terms of a higher level of knowledge, the event assimilator would step in. However, this approach posed a fundamental problem for parser unification.

A unified parser cannot rely upon an event assimilation module to aid in the parse for the following simple reason: Once the narrative has been completely processed, event assimilation expectations no longer exist. These expectations arose as each sentence in the narrative was being read. When the comprehension task is complete and the entire narrative has been processed, there are no more active expectations. But the description of any event which occurred in a narrative may appear within a subsequent question about that narrative. Thus, the QA parser has to be able to parse event descriptions which were parsed at story-understanding time, but without the help of an event assimilation module. It is important to have a QA parser which can work after all active story expectations have terminated.

In fact, the question answering parser (DYPAR) had been parsing event descriptions successfully without relying on an event assimilation module. As we have already stated, it accomplished this by searching episodic memory during the parse. Thus, the solution seemed clear: unify the two parsers both by eliminating the event assimilator and by augmenting the QA parser to build episodic memory during its parsing process. This was accomplished, yielding a completely integrated system in which there is a single parser scheduling all processing for both story understanding and question answering.

Specifically, this complete integration was achieved in the following way: Unlike FRUMP and other Hop-down'processors, BORIS operates in a very 'bottom-up' manner. Active processes called "demons" are associated with lexical and phrasal entities. In addition to syntactic and semantic sentence analysis, these demons also search and construct episodic memory.

Although these demons can not be expected to know about the specific content of episodic memory, they do know enough about the structural principles of episodic memory to search it and determine how each lexical entity should be interpreted. This is done by finding an appropriate knowledge structure in memory, and activating processes associated with that knowledge structure. It is memory search demons associated with lexical items that ultimately determine which knowledge structure to apply.

> **Memory access and parsing are two sides of the same coin: Parsing organizes memory access.**

To get an idea of how parsing can entail access to episodic memory, consider the following short story, with two alternative endings:

> Bill was very jealous of his wife Mary. He decided to hire George, a private detective, to keep an eye on her. George took his miniature camera with him to the school where Mary worked.
>
> FI: There, George found Mary with another teacher.
>
> E2. There, George found Mary with another man.

In EI, the inference is that Mary is also a teacher. However, the inference in E2 is not that Mary is also a man. Instead, the inference to be made in E2 is that Mary is with a man other than her husband Bill (versus 'other than' George).

In addition, if E1 had been read in isolation, it would not be clear whether it is George or Mary who is also a teacher. Interpretive inferences in these cases depend on what immediately precedes and follows "another", and the relationship between George and Mary. The moment "another" finds the object.it is modifying, it activates demons which search episodic memory for the needed interpersonal information.[1]

3.1. Event Explanation

Integration in BORIS has resulted in a system which is more explanation-based than expectation-based. That is, each input must search memory in order to explain itself, instead of relying upon some "top-down" process to grab it and interpret it. In BORIS, most expectations [12] are encoded implicitly within the structure of episodic memory, rather than being active processes which poll the input. Because expectations in BORIS fall into a fixed number of classes, each class can be represented declaratively in terms of the kinds of memories which are being constructed.

For example, when BORIS reads:

S1: The teacher examined the student,

"examined" is represented as:

```
EV-1: (D-KNOW ACTOR x
              OBJECT (STATE OF y)
              INSTRU (ATTEND TO y))
```

BORIS tries to 'explain' EV-1 in the following way:

> To explain an event E, use whatever bottom-up knowledge is associated with the event to decide where in memory to search. Once a knowledge structure (KS) is found, apply whatever processing is associated with that KS to E. This may result in building new structures (or connections) in episodic memory.

For instance, BORIS knows that the object bound to the ACTOR slot in EV-1 could tell it where to search. Since x is bound to RT-TEACHER (role theme information about teachers), BORIS searches knowledge structures associated with RT-TEACHER. Within RT-TEACHER are pointers to other memory structures (GIVE-TEST in M-EDUCATION), which will match the description of EV-I. Thus, EV-1 will be 'explained* in terms of M-EDUCATION. This particular search heuristic will also work correctly in other cases, such as:

S2: The doctor examined the student.

which refers to a rather different situation than the one described byS1.

When the semantics of the input sentence fails to constrain the memory search, episodic memory is searched according to recency and the first structure which matches the input is then used to interpret the input. This is similar in spirit to having explicit expectations each poll the input according to recency. Thus, the explanation-based approach compares favorably with having expectation-driven techniques.

When is an event fully 'explained? This is a difficult issue which ultimately depends on the beliefs and interests that a reader brings to the subject matter being read. A satisfactory explanation for one person may seem superficial to another. The intelligence of a reader depends on the depth of explanation sought and what is appropriate given the surrounding circumstances.

In earlier versions of BORIS, two different approaches were tried:

(a) Once one knowledge structure has been found which matches the input, then the input has been explained.

(b) Exhaustively apply every knowledge structure to the input.

But there are weaknesses with each of these strategies. The first approach is inadequate for an understanding system which supports multiple perspectives [9] because the explanation process will terminate after only one perspective is found. For example, one narrative BORIS reads is DIVORCE! In it, two characters (Richard and Paul) agree to meet at lunch for the following reasons: a) they haven't seen each other for years, and b) Richard is a lawyer and Paul wants Richard to represent him in the divorce case. Their meeting, therefore, has more than one perspective:

Scriptal: restaurant meal occurring.

Thematic: suspended friendship being renewed.

Role: lawyer and client meeting to discuss legal case.

The second approach solves the multiple perspective problem by examining every active knowledge structure in order to explain a given input. However, this creates a very unfocussed system, which will always be considering lots of irrelevant knowledge structures. This approach, therefore, is inherently inefficient.

The current approach used in BORIS is to analyze an event at several specified levels - scriptal, goal/plan, thematic, and

[1] The word "another" appears in two narratives BORIS has processed, in one, BORIS infers that a character is a teacher. In the other BORIS infers marital infidelity. For the text of these narratives, see [9] [7],

role. Once an event has been analyzed in terms of each level, the processing for that event terminates.

3.2. Parsing in Different Modes

Although both narrative sentences and questions are parsed by the same program, there are several situations in which the "mode" of understanding (question answering versus story understanding) requires that the parser behave differently:

3.2.1. Tokentzotkm

Whenever the parser encounters a reference to an object primitive, a setting, or a character, memory is searched to find out if the object already exists in episodic memory. In story-understanding mode, the failure to find a referent results in the creation of a new token in memory. However, during question answering, a failure to find a referent results in abandoning any attempt to find an answer. Instead, a 'complaint' is generated to point out that a referent could not be found. For example, given a story only about two characters. Richard and Paul, BORIS would respond as follows:

Q: What does George do for a living?
A: I don't recall any mention of a person named George in the story.

3.2.2. Presupposition Cheeking

During story understanding, if a knowledge structure in episodic memory is referenced and the role bindings created during the parse fail to match the bindings already in memory, this indicates that a new instantiation must be built. For example, in DIVORCE-!, Richard receives a letter from Paul. If a letter were subsequently mentioned as having been written by another character (say Sarah), then BORIS would create a new instance of a letter knowledge structure in memory.

During question answering, the parser also checks the role bindings presumed by the question against the actual bindings maintained in episodic memory. In this case, however, a failure of roles to match indicates a false presupposition. So BORIS rejects the question and corrects the false presupposition. For instance:

Q: Why did Sarah write to Richard?
A: It was Paul, not Sarah, who wrote to Richard.

People sometimes fail to notice false presuppositions. In such cases the presupposition can become incorporated into their memory model of the narrative (see section 4).

3.2.3. Question Words

In any narrative or question, words like "who", "what", "why" , "where", etc. must be handled in special ways. If such a word is encountered during question answering, it is assumed to be initiating a request for information. During story understanding however, different demons process these words when they occur at clause boundaries. So if a question is encountered while the story is being comprehended, it will be treated as a rhetorical question, and no attempt will be made to answer it.

3.2.4. Loco! Contexts

Another difference between story understanding and question answering modes arises from the use of local contexts in story understanding and question answering tasks. Local contexts arc represented by the memory created for events immediately prior to some event currently being explained. Consider the following sentence, (from [9]

Richard had borrowed money which was never puid back, but now he didn't know where to find his old friend.

The parse of the first clause causes the knowledge structure M-BORROW to be built, within which the event RETURN-OBJ is marked as violated. The second clause, however, is understood with the help of a local M-BORROW context. As the parser produces a representation of Richard's goal failure (not knowing where to find Paul), the following 'negative event* rule is activated:

If a negative event E2 occurs,
and local context contains a
violation of a prior event E_1
Then see if E2 enables E1. and
if so, create link: <E 1 blocked-by E2>
as an explanation for the violation.

Thus, 'not knowing where to find Paul' is understood as the explanation for 'never having paid Paul back' by using the above rule, along with enablement information supplied by M-BORROW

A different kind of local context exists during question answering. In this case, the local context contains the last question asked, along with its answer.[2] Consider the following question and its alternative answers:

Q: Did Richard know where to find his old friend?

A1: Yes. they met at a restaurant.
A2: Yes. Paul had written to Richard.
A3: No. Richard hadn't heard from Paul in years.

Although this question contains information similar to its narrative counterpart. BORIS will find answers Al and A2 before A3 because the local context provide constraints differing from those during story understanding.

More specifically, there is no direct access to all of Richard's knowledge states in BORIS. Instead. BORIS is forced to search a scenario map [4] in order to see if Paul and Richard ever interacted with one another. However, understanding the question:

Q: Why hadn't Richard paid Paul back?

accesses M-BORROW directly (as it was accessed at story understanding time). Next the <blocked-by> link, which was constructed during the comprehension phase, is retrieved by an

[2] One use of local context during question answering is to handle such situations as:

Q1: Who wrote Richard a letter? A1: Paul.
02: Why?

where both the previous question and answer are needed to understand a subsequent question.

interference search [4]. This results in the answer:

A: Richard didn't know where Paul lived.

One natural consequence of the different uses of local context in story understanding and question answering is that it is difficult to understand a question about an event which required a local context when being understood at story understanding time.

4. Memory Modification During QA

As parsing became unified and integrated in BORIS, an unanticipated situation arose during question answering: asking BORIS certain questions created episodic memory modifications. At first we thought this effect was a 'bug' in the program, and the difference in parsing modes (see section 3.1.) was set up in an attempt to eliminate it.

But on further reflection, there is no reason to believe that people only construct memories during narrative comprehension, any more than they only search memories during question answering. In fact, psychological experiments by Elizabeth Loftus [II] have demonstrated that the process of asking questions can modify the memory of eyewitnesses. Preliminary experiments performed at Yale [8] show that the Loftus effect occurs in the case of narratives as well. It follows that memory modification during question answering is a natural consequence of having a unified and integrated parser.

How do memory modifications during question answering actually occur? If we assume that the mode of processing (as described in section 3.1) determines when new memories can be built, then memory modification depends on "fooling" an understanding system in question answering mode to think that it is operating in story understanding mode. This naturally occurs if the understander assumes that whoever is asking a question knows something about the narrative and may impart this information within the question. Questions normally convey information by means of presuppositions. For example, suppose we read a story about John going to a store, and then we are asked:

Q: When did John go to buy groceries?

Here, the questioner presupposes that John went specifically to a grocery store. If we believe that the questioner knows something about John's activities, then we will accept being told new information in the question. This is similar to having heard originally that John went grocery shopping. So modification of old information during question answering depends upon:

(1) An acceptance (by the answerer) that the questioner may know something about the situation under examination.

(2) New information supplied in the presupposition of the question.

In BORIS, memory modification can occur because the processes which add information to an event description are the same processes which check presuppositions. For example, if a narrative sentence states "John got a letter from Mary" then a letter-event must be instantiated with the bindings: <sendcr = Mary> and <receiver = John>. In order to determine that this is not a reference to a new letter-event, memory search processes must search to see if another letter-event exists with different bindings. Now suppose that BORIS is asked: "Why did John get a letter from Mary?". Again we must search memory for a letter-event which shares these bindings. So presupposition checking and role-binding instantiations are performed at the same point during both story understanding and question answering.

However, if a question presupposition clearly violates our memory, we will complain. So the question: "Why did John kill Mary?" — in a story about John and Mary getting married - will not be accepted.

The experiments in [8] indicate that memory modifications during question answering are more likely to occur when the presuppositions in the question do not violate any important script, plan or goal information in the narrative. Modifications during question answering depend on what information was built in memory during narrative comprehension, and how strongly presuppositions conflict with it. If absolutely everything stated (or inferable) in a narrative were explicitly instantiated, then presupposition checking would always notice new information assumed in a question. However, if memory is largely reconstructed from more general information, then it is difficult for a presupposition checker to distinguish new information from old - as long as the new does not violate anything on purely reconstructive grounds.

There is good evidence that human memory is largely reconstructive. For example. Spiro et al. [14] used the following experimental materials in a series of reading experiments:

(a) The karate champion hit the block.

(b) The block broke.

(c) He had had a fight with his wife earlier. It was impairing his concentration.

The following claims were made about reconstructive memory:

(1) If only (a) and (b) were first presented, then (b) could be reconstructed from (a), so (b) would not be instantiated in memory. Therefore, a later presentation of (c) would cause people to falsely assume that (b) had not occurred, and memory of (b) would be degraded.

(2) If (c) were presented before (a) and (b), then (b) could not be easily reconstructed. Therefore, (b) would be explicitly instantiated, so memory of (b) would not be degraded.

These claims were supported by the experimental data. Whenever people could infer the consequence of an antecedent they did not have to instantiate it, since it was reconstructable. This meant that false information about reconstructable events is more likely to be integrated into an existing memory representation.

5. *Memory Reconstruction in BORIS*

In keeping with the reconstructive claims espoused in Spiro et al. [14], the BORIS system tries to follow a simple principle:

Don't instantiate anything which can be reconstructed at a later date.

Whenever a knowledge structure is successfully applit in interpreting an input, a memory instantiation is created (or updated), in which the following information is stored:

1. A pointer to the knowledge structure which was activated (If we dont instantiate this, BORIS wouldn't know what had happened versus what had not happened.)

2. Any role bindings (so that BORIS can recall who did what to whom).

3. Scenario information (so that an event can be placed in a spatio-temporal relationship with other events)

4. "How far" along in the knowledge structure BORIS has progressed.

5. Any higher level goals which this knowledge structure has either achieved, or intends to achieve.

6. Any violations or deviations of this knowledge structure.

Therefore, any event central to a script, scenario, or goal is instantiated. By definition, violations and deviations can not be reconstructed, so they are instantiated also. For instance, when BORIS reads that a waitress spills coffee on someone, then the spill-event must be explicitly instantiated both as a violation of BRING-FOOD in RESTAURANT and a violation of DO-SERVICE in SERVICE. For more discussion of violation handling in BORIS, see [6] and [7].

6. Conclusions

In a completely integrated system, both parsing and memory processes require a fresh examination. Our experience suggests that one way to proceed *is* to build fully integrated systems in which a single parsing process invokes all comprehension tasks on an on-going basis.

This article has discussed some consequences arising from unifying parsing for both question answering and narrative comprehension in an integrated system such as BORIS. Two major consequences are:

- Answers to questions may be known before questions have been completely understood.
- Asking questions may modify episodic memory by augmenting it with new information contained within the questions.

These phenomena resemble the type of behavior people exhibit when reading and answering questions about narratives, and deserve further experimentation in both the computational environment and the psychology laboratory.

Acknowledgements

Special acknowledgements go both to Margot Flowers, and to my thesis advisor, Wendy Lehnert, for their valuable suggestions and edits on earlier drafts of this paper. Special (hanks go also to Tom Wolf, who has been central in the current design and implementation of BORIS. Finally, my appreciation goes to Pete Johnson. Mark Burstein, Marty Korsin. CJ Yang and Steve Harley for their past contributions to the BORIS project.

References

[1] **Birnbaum L. and M. Selfridge. "Conceptual Analysis of Natural Language." in Inside Computer Understanding: Five Programs Plus Miniatures R. Schank and C. Riesbeck (eds). Lawrence Erlbaum NJ. 1981.**

[2] Cullingford, R. E. "Script Application: Computer Understanding of Newspaper Stories." Tech. Rep. 116. Yale University. Computer Science Dept. 1978.

[3] DeJong II. G F "Skimming Stories in Real Time: An Experiment in Integrated Understanding." Tech. Rep. 158, Yale University. Computer Science Dept. 1979.

[4] Dyer, Michael G.and W. G. Lehnert. "Organization and Search Processes for Narratives." Tech. Rep. 175, Yale University. Computer Science Dept. April 1980.

[5] Dyer, Michael G. "DYPAR - A Demon-Based Parser." In W. G. Lehnert. Processes of Thought and Language. (In preparation) 1981.

[6] Dyer. Michael G "The Role of TAUs in Narratives." Proc. of Third Annual Cognitive Science Conference. At Univ. of California. Berkeley. August 1981.

[7] Dyer. Michael G. In-Depth Understanding. A Computer Model of Memory for Narrative Comprehension. PhD Thesis, Computer Science Dept. Yale Univ. (forthcoming). 1981.

[8] Lehnert, W. G. and Robertson, S. P. "Memory Modification During Question Answering." Text Comprehension Symposium at the Deutsches Institut fur Fernstudien an der Universitat Tubingen. West Germany. 1981.

[9] Lehnert, W G., Dyer. M. G., Johnson. P. N.. Yang, C.J.. and S. Harley. "BORIS - An Experiment in In-Depth Understanding of Narratives." Tech. Rep. 188. Yale Univ. Computer Science Dept. December 1980.

[10] Lehnert. Wendy G **The Process of Question Answering.** Lawrence Erlbaum. Hillsdale. NJ. 1978.

[11] Loftus. E. F. Eyewitness Testimony. Harvard Univ. Press, Cambridge, MA. 1979.

[12] Riesbeck, C and R. Schank. "Comprehension by Computer: Expectation-Based Analysis of Sentences in Context." Tech. Rep. 78. Yale University. Computer Science Dept. 1976.

[13] Schank. Rand Abelson, R. Scripts, Plans, Goals, and Understanding. Lawrence Earlbaum, Hillsdale, NJ. 1977.

[14] Spiro. R.J., Esposito, J.and R.J. Vondruska. "The Representation of Derivable Information in Memory. " In TINLAP-2: **Theoretical Issues in Natural Language** Processing. Univ. of Illinois at Urbana-Champaign, 1978.

[15] Wilensky, R. "Understanding Goal-Based Stories," Tech. Rep. 140. Yale Univ. Computer Science Dept. 1978.