A Parallel-Process Model of On-Line Inference Processing

Kurt P. Eiselt

Irvine Computational Intelligence Project
Computer Science Department
University of California
Irvine, California 92717

Abstract

This paper presents a new model of on-line inference processes during text understanding. The model, called ATLAST, integrates inference processing at the lexical, syntactic, and pragmatic levels of understanding, and is consistent with the results of controlled psychological experiments. ATLAST interprets input text through the interaction of independent but communicating inference processes running in parallel. The focus of this paper is on the initial computer implementation of the ATLAST model, and some observations and issues which arise from that implementation.

1.0 Introduction

This paper describes a new theory of inference processing developed at the Irvine Computational Intelligence Project, and an initial computer implementation of that theory. The research described here integrates inference processing at the lexical, syntactic, and pragmatic levels, and is consistent with the results of controlled psychological experiments. The theory centers upon a parallel-process model of text understanding which explains inference behavior at the different levels as the result of interactions between three independent but communicating inference processes. Though there are three processes operating at three different levels of language understanding, there is no direct correspondence between the levels and the processes. Inference decisions at all levels are made through the combined actions of the three processes running in parallel. We call this model ATLAST (A Three-level Language Analysis SysTem).

ATLAST represents a real departure from most previous models of language understanding and inference processing [e.g., Schank, 1975; Cullingford, 1978; Wilensky, 1978; DeJong, 1979], though there are models which integrate some of the levels of inference processing. For example, IPP [Lebowitz, 1980] and BORIS [Dyer, 1982] integrate the syntactic and pragmatic levels, while the model of Small, Cottrel), and Shastri [1982] integrates lexical access and syntactic parsing. Finally, Charniak's model, as does ATLAST, seeks to integrate lexical, syntactic, and pragmatic inference processing [Chamiak, 1983], though his model differs from ATLAST in other respects.

2.0 Background: The Theory in Brief

The theory behind ATLAST is described in detail in [Granger, Eiselt, & Holbrook, 1985], but a brief review of the theory is provided here to aid in understanding the program.

ATLAST is a direct descendant of earlier work on inference decision processes at the pragmatic level. Specifically, it came about as an attempt to address word-sense ambiguity problems which arose during research into different pragmatic inference strategies used by human subjects while reading text, and the development of a program, called STRATEGIST, which modelled that behavior [Granger, Eiselt, & Holbrook, 1983; Granger & Holbrook, 1983). As we worked on STRATEGIST, we observed that lexical and pragmatic inference processes appeared to have much in common. Many pragmatic inferences seemed to be triggered by individual words. This is hardly new news, of course, as there exist integrated models of language understanding in which higher-level inferences are directly activated by input text (FRUMP [DeJong, 1979] and IPP [Lebowitz, 1980] are notable examples). We believed, though, that the relationship was even closer than described by previous models that the inference decision mechanisms themselves were in some way interdependent at the very least. For example, in the text

The CIA called in an inspector to check for bugs.
The secretaries had reported seeing roaches.

the first sentence alone has an unambiguous interpretation: the "hidden microphone" sense of "bugs" is more appropriate than the "insect" sense. Upon reading the second sentence, the "insect" sense is obviously more appropriate, and the initial choice of word-sense for "bugs" must be supplanted. To explain this process, we theorize that "CIA" triggers a pragmatic inference about espionage which in turn influences the choice of the "hidden microphone" word-sense for "bugs". Later, the word "roaches" generates higher-level inferences which suggest that the "insect" sense of "bugs" is correct and that the Central Intelligence Agency, in this case at least, is more appropriately

This research was supported in part by the National Science Foundation under grant IST-81-20685 and by the Naval Ocean Systems Center under contracts N00123-81-C-1078 and NC6001-83-C-0256.

viewed as a generic employer trying to rid itself of pests rather than as an espionage agency protecting its secrets. Thus, the context generated by "CIA" determined the selection of a word-sense? for "bugs", while the context generated by "roaches" resulted in an entirely new interpretation of "bugs" and a slightly modified interpretation of "CIA". Because of this interdependence between inference levels, theories about pragmatic inference mechanisms must include theories about lexical access processes.

Lexical access is the process by which a word's meaning is extracted from its written (or spoken) form. Recent research into lexical access has led to the counter-intuitive conclusion that when an ambiguous word is presented in context (i.e., a sentence or phrase), all meanings of the word are initially accessed, and context is subsequently consulted to determine the most appropriate meaning [Swinney & Hakes, 197G; Tanenhaus, Leiman, & Seidenberg, 1979; Lucas, 1983; Granger, Holbrook, & Eiselt, 1984]. This happens regardless of the syntactic category of the word, or whether the context is biased toward one meaning or another.

If the lexical access process does in fact work as described above, and if individual words trigger the higher-level pragmatic inferences, then it is likely that the pragmatic inference decision process is much the same as the lexical inference decision process. Work on ATLAST goes under the assumption that, when more than one interpretation (i.e., pragmatic inference) of an input text is possible, all possible interpretations are pursued in parallel, and those interpretations which do not fit well with the existing context are "de-activated" or inhibited.

3.0 How ATLAST Works

3.1 Memory

ATLAST is built around a high-level episodic memory structure which contains two kinds of memory organization packets (MOPs) [Schank, 1982; Kolodner, 1984]. For each word in ATLAST's vocabulary there is a MOP which represents that word. Most lexical-entry MOPs contain a one-way link to one or more word-senses directly associated with that word, and syntactic information about the word-senses. Function words, such as "a" and "the", are not linked to other MOPs and serve only to aid in syntactic decisions. The word-senses are an example of the other kind of MOP in ATLAST\s memory: those which represent events or objects. These MOPs are interconnected through a network of two-way links which serve to define the relationships between the MOPs. These MOPs can be, but are not necessarily, directly linked to lexical entries.

The inference decisions in ATLAST are carried out by three primary components: the Capsulizer, the Proposer, and the Filter. Theoretically, these processes run in parallel. However, ATLAST is written in UCI-LISP on a DECSYSTEM-20, so the parallelism which is so important to the theory is necessarily simulated in its implementation. This simulation is accomplished by repeatedly cycling through the three processes. Thus, the Capsulizer runs for a pre-determined amount of time, followed by the Proposer, then the Filter, then the Capsulizer again, and so on. The amount of time each process is allocated is an important issue with respect to the accuracy of the model. This issue has not yet been fully explored.

3.2 Capsulizer

The Capsulizer contains the first stage of a two-stage syntactic analysis process similar in some respects to that described by Frazier and Fodor [1978]. The Capsulizer makes intra-phrasal, as opposed to inter-phrasal, syntactic decisions about the words in the input text (again, see [Granger, Eiselt, & Holbrook, 1985] for a discussion of the theory behind two-stage syntactic analysis). As the Capsulizer encounters each new word in the input text, it retrieves the syntactic category information associated with that word (e.g., "this word can be used as a noun and a verb") and activates any word-senses associated with that word. The word-senses are not used in any decisions made by the Capsulizer, though pointers to the word-senses are retained. The activated word-senses serve as a starting point for the search carried out by the Proposer, which is described below.

As the Capsulizer processes the input words, it accumulates the syntactic information it retrieves and makes initial decisions about syntactic relationships within the phrases of the input text. These intra-phrasal decisions, along with the pointers to the word-senses which comprise the phrases, are passed along to the Filter as "capsules" of information. The Filter then makes decisions about the syntactic relationships between the phrases (i.e., inter-phrasal syntax). If an input word activates more than one word-sense (i.e., a word-sense ambiguity), the pointers to the multiple word-senses are all passed on to the Filter, which will eventually select the "best" word-sense. This process is also described in more detail below.

3 3 Proposer

The Proposer gets its name from the idea that it "proposes" possible inference paths which might explain the input text. Essentially, it is a search mechanism which employs spreading activation to traverse the links between the MOPs in memory and find connections between word-senses which have been activated by the Capsulizer.

The Proposer maintains pointers to the most recently activated MOPs in memory, and to the word-senses which are the origins of the spreading activation search. Each time the Proposer is invoked, it traverses the links leading away from the recently activated MOPs, activates the adjacent MOPs at the end of those links, and updates its list of pointers. If the spread of activation from one point of origin intersects the spread of activation from some other point of origin, then the Proposer has found some plausible relationship, by way of links and MOPs, between two (and possibly more) of the word-senses activated by the input text. The Proposer then passes information about this newly-discovered pathway to the Filter; in this way, the Proposer "proposes" possible inference paths for evaluation by the Filter.

Spreading activation has been employed in a number of models [e.g., Quillian, 1968; Fahlman, 1979; Anderson, 1983; Charniak, 1983; Norvig, 1983; Riesbeck *k* Martin, 1985]. Spreading activation allows ATLAST to pursue multiple inference paths in parallel. Were this process allowed to continue unchecked, it would lead to a combinatorial explosion of inference paths. To prevent this from happening in ATLAST, the third major process, the Filter, constantly evaluates or "filters" inference paths and inhibits pursuit of those which appear to be poor explanations of the input text. Though the idea of beginning pursuit on all inference paths instead of just the "appropriate" ones may seem both counter-intuitive and counter-productive, there are two arguments for using this approach. One is that it would seem impossible to determine which inferences may be appropriate without first evaluating all inference possibilities. The other is that this approach is consistent with experimental studies of human behavior [Tanenhaus, Leiman, & Seidenberg, 1979; Seidenberg, Tanenhaus, lyeiman, *k* Bienkowski, 1982; Granger, Holbrook, & Eiselt, 1984].

The Proposer is implemented in ATLAST as a separate process, but from a theoretical perspective it might be more appropriately viewed as an emergent property of a human memory organization. Computer memory seems to work somewhat differently than human memory, though, so it was necessary to provide a separate process to make the spreading activation possible.

3 4 Filter

The Filter performs two functions; the first is that of inter-phrasal syntax. As capsules are passed from the Capsulizer to the Filter, the Filter makes decisions about the relationships between the phrases represented by the capsules. Inter-phrasal syntax rules enable the Filter to fill the Actor, Action, and Object slots, for example. Future work on the ATLAST program will add rules about modifying phrases, keeping track of referents across phrases, and agreement of tense, number, and gender, among other rules.

The Filter's other function is the evaluation of inference paths. When two competing inference paths are proposed (e.g., different paths connecting the word-senses of two words from the input text), the Filter attempts to select the more appropriate path through the application of three inference evaluation metrics.

First, the Filter evaluates the inference paths according to the specificity metric [Wilensky, 1983]. If one path is determined to be less specific than the other, the less specific path is inhibited; that is, the spread of activation from nodes on the path is stopped, and that path is no longer considered as a plausible explanation for the input text. Specificity is determined by the links in the path: a path which includes a "viewed-as" link (from the "view" relationship defined in [Wilensky, 1984]) is less specific than a path which does not contain such a link. In the example of Section 3.5, the CIA is a special case of a spy agency, but a spy agency can also be viewed as an employer; an inference path which describes the CIA only as a spy agency is more specific than one which explains it as a spy agency and an employer.

If the specificity metric fails to make a decision between two competing paths, the Filter applies two variations of the parsimony metric [Granger, 1980]. The first of these variations (the "length" metric) gives precedence to the inference path with fewer links. Failing this, the Filter applies the other variation of the parsimony metric (the "explains more" metric), which examines the "intermediate" MOPs (those which are not the endpoints) of the two competing paths. This variation then selects the inference path that contains more intermediate MOPs which are intersection points with other inference paths. In other words, the intermediate MOPs can be either endpoints or intermediate MOPs of paths other than the two being evaluated.

It is with the Filter that the implementation of ATLAST diverges most from the theory. In some sense, this is to be expected, since the Filter is the most complex of the three processes. In theory, ATLAST should be able to evaluate and inhibit pursuit of apparently implausible inference paths almost as soon as pursuit has begun, thus preventing problems of combinatorial explosion. ATLAST would accomplish this by comparing the multiple, possibly incomplete, inference paths which begin with a specific word-sense to the context it has built up to that point in the processing of the input text, and determining which of the paths fit "best" with that context. This would be in agreement, with experimental results in lexical access research [Tanenhaus, Leiman, & Seidenberg, 1979; Lucas, 1983; Granger, Holbrook, & Eiselt, 1984]. At this time, the ATLAST model can only evaluate complete inference paths (i.e., those which connect two or more word-senses activated by the Capsulizer) without regard to the existing context. Though this simple inference evaluation mechanism seems to work for sentences such as the one presented in the following example, it will not be sufficient to properly interpret longer, more complicated texts. This problem will be rectified in the near future.

3.5 An Example

What follows is actual (though abbreviated) annotated run-time output from the ATLAST prototype program. This example illustrates primarily how ATLAST disambiguates between two possible meanings of the word "bugs" in the text, "The CIA checked for bugs." In the interest of brevity and clarity, we use a very short text and just enough of a knowledge base to process this example. Due to space limitations, we will concentrate primarily on the operation of the Filter. Also, we have abbreviated the names of some of the memory structures, again due to space limitations. The following legend should make the program trace more readable:

GEN-EMPLOYER - GENERIC-EMPLOYER
GET-SECRETS - GET-OTHERS-SECRETS
MPHONE = MICROPHONE
P-HEALTHY-ENVT - PRESERVE-HEALTHY-ENVIRONMENT
P-SECRETS - PRESERVE-OWN-SECRETS
PLANT-LISTEN-DEV = PLANT-OWN-LISTENING-DEVICE
REM-HEALTH-HZRD « REMOVE-HEALTH-HAZARD
REM-LISTEN-DEV - REMOVE-OTHERS-LISTENING-DEVICE

After processing "The CIA" and activating associated memory structures, ATLAST processes "checked", which terminates the noun phrase and begins a verb phrase. Capsulizer sends a capsule consisting of the word-senses initially activated by the noun phrase (i.e., C-I-A) to Filter. Filter, looking for an actor for this sentence, fills the slot with this noun-phrase capsule. ATLAST then processes "for":

```
Filter:
  New path discovered: IPATHO
    Path from C-I-A to SEARCH
      C-I-A is special case of SPY-AGENCY
      SPY-AGENCY has goal P-SECRETS
      P-SECRETS has plan REM-LISTEN-DEV
      REM-LISTEN-DEV is special case of REMOVE
      REMOVE has precondition SEARCH
  ACTION slot filled by SEARCH
```

The preposition "for" does not activate any new memory structures, but it does begin a modifying prepositional phrase. Capsulizer sends the verb component of the verb phrase (SEARCH) to Filter, which then assigns the capsule to the action slot.

Proposer, looking for intersections among the "wavefronts" of spreading activation, finds a connection, or inference path (IPATHO), between C-I-A and SEARCH, and notifies Filter. Filter knows of only one inference path at this time, so there is no basis for comparison and evaluation of inference paths yet. ATLAST then moves on to "bugs":

```
Filter.
  New path discovered: IPATH1
    Path from C-I-A to SEARCH
      C-I-A is special case of SPY-AGENCY
      SPY-AGENCY can be viewed as GEN-EMPLOYER
      GEN-EMPLOYER has goal P-HEALTHY-ENVT
      P-HEALTHY-ENVT has plan REM-HEALTH-HZRD
      REM-HEALTH-HZRD is special case of REMOVE
      REMOVE has precondition SEARCH
  New path discovered. IPATH2
    Path from C-I-A to INSECT
      C-I-A is special case of SPY-AGENCY
      SPY-AGENCY can be viewed as GEN-EMPLOYER
      GEN-EMPLOYER has goal P-HEALTHY-ENVT
      P-HEALTHY-ENVT has plan REM-HEALTH-HZRD
      REM-HEALTH-HZRD has role-filler INSECT
  New path discovered: IPATH3
    Path from C-I-A to MPHONE
      C-I-A is special case of SPY-AGENCY
      SPY-AGENCY has goal GET-SECRETS
      GET-SECRETS has plan PLANT-LISTEN-DEV
      PLANT-LISTEN-DEV has role-filler MPHONE
  New path discovered: IPATH4
    Path from C-I-A to SEARCH
      C-I-A is special case of SPY-AGENCY
      SPY-AGENCY has goal GET-SECRETS
      GET-SECRETS has plan PLANT-LISTEN-DEV
      PLANT-LISTEN-DEV has role-filler MPHONE
      MPHONE is role-filler of REM-LISTEN-DEV
      REM-LISTEN-DEV is special case of REMOVE
      REMOVE has precondition SEARCH
  New path discovered: IPATH5
    Path from SEARCH to MPHONE
      SEARCH is precondition of REMOVE
      REMOVE has special case REM-LISTEN-DEV
      REM-LISTEN-DEV has role-filler MPHONE
  New path discovered: IPATH6
    Path from SEARCH to INSECT
      SEARCH is precondition of REMOVE
      REMOVE has special case REM-HEALTH-HZRD
      REM-HEALTH-HZRD has role-filler INSECT
  New path discovered: IPATH7
    Path from MPHONE to C-I-A
      MPHONE is role-filler of REM-LISTEN-DEV
      REM-LISTEN-DEV is plan of P-SECRETS
      P-SECRETS is goal of SPY-AGENCY
      SPY-AGENCY has special case C-I-A
  Parsimony: IPATH7 explains more than IPATH3
  Specificity: IPATH4 more specific than IPATH1
  Parsimony: IPATHO shorter than IPATH4
```

Capsulizer reads the ambiguous word "bugs", which results in the activation of two word-senses: INSECT and MPHONE. Proposer's search has uncovered several new inference paths. When two different inference paths connect the same two word-senses, Filter applies inference evaluation metrics to the two paths to determine which of the two provides the better explanation of the input text. The rejected paths are de-activated until later text results in activating that path again. Finally, ATLAST encounters the end of the text:

```
Filter:
  OBJECT has competing slot fillers:
    INSECT vs. MPHONE
  Specificity: IPATH7 more specific than IPATH2
  Parsimony: IPATH5 explains more than IPATHO
  Lexical ambiguity resolution:
    MPHONE vs. INSECT
```

```
   All paths through INSECT de-activated
   Ambiguity resolved: MPHONE selected
 OBJECT slot filled by MPHONE
```

Capsulizer sends to Filter a capsule containing the word-senses activated by the prepositional phrase. Filter determines that the capsule contains the object of the action SEARCH, and that this object is ambiguous. Filter attempts to resolve this ambiguity by applying the inference evaluation metrics to the remaining active inference paths. Because MPHONE and INSECT are now known to be competing word-senses, Filter treats IPATH7 and IPATH2 as competing inference paths. That is, although IPATH7 connects MPHONE to C-I-A and IPATH2 connects INSECT to C-I-A, the two different paths are evaluated as if they connected the same two word-senses because INSECT and MPHONE were activated by the same lexical entry ("bugs"). For this same reason, IPATH5 is evaluated against IPATH6. This evaluation results in the two remaining inference paths containing INSECT to be de-activated, so Filter resolves the ambiguity in favor of MPHONE. Below is the active memory structure after all processing has ended, followed by the pointers into the structure.

```
Processing completed
  Active memory structure:
    Path from MPHONE to C-I-A
      MPHONE is role-filler of REM-LISTEN-DEV
      REM-LISTEN-DEV is plan of P-SECRETS
      P-SECRETS is goal of SPY-AGENCY
      SPY-AGENCY has special case C-I-A
    Path from SEARCH to MPHONE
      SEARCH is precondition of REMOVE
      REMOVE has special case REM-LISTEN-DEV
      REM-LISTEN-DEV has role-filler MPHONE
    Path from C-I-A to SEARCH
      C-I-A is special case of SPY-AGENCY
      SPY-AGENCY has goal P-SECRETS
      P-SECRETS has plan REM-LISTEN-DEV
      REM-LISTEN-DEV is special case of REMOVE
      REMOVE has precondition SEARCH
  Pointers to memory structure:
    Actor: C-I-A
    Action: SEARCH
    Object: MPHONE
```

3.6 An Observation on the Ordering of Inference Metrics

While testing the ATLAST program, it became apparent that the order of application of the pragmatic inference metrics affected ATLAST's eventual interpretation of the input text. As mentioned earlier, ATLAST applies its specificity metric first, followed by the "length" metric, and then the "explains more" metric. For the example of Section 3.5, this ordering of the inference metrics results in the interpretation that the CIA was looking for hidden microphones. On the other hand, if the order of application of the two parsimony metrics is reversed, ATLAST arrives at a different, nonsensical interpretation.

Though this observation does not lead us to any meaningful conclusions at this time, it provides an example of how ATLAST can serve not only as a "proving ground" for theories, but also as a source of new and interesting ideas worthy of further investigation.

4.0 Open Questions and Future Work

The initial implementation of ATLAST raised a myriad of implementation issues, many of which are yet to be resolved. More importantly, the implementation again raised some open questions which have been encountered by other researchers.

One question has to do with the timing of the three inference processes running in parallel. We do not yet know how much work each of the three processes should do in a given cycle, though we have made arbitrary initial decisions. For the Proposer in particular, there are issues which have been addressed by some of the previous models utilizing spreading activation [Quillian, 1968; Fahlman, 1979; Anderson, 1983]: How far does activation spread? Does activation decay with time? Is there reinforcement when paths intersect? Though we do not have answers to the questions now, ATLAST is designed to allow us to change timing parameters easily, possibly enabling us to "tune" the model for cognitive accuracy as work proceeds.

Another question is concerned with the content of ATLAST's memory. Currently, ATLAST runs with a high-level abstraction of episodic memory: the relationships between the MOPs are fairly well defined, but the details of the episodes themselves are almost non-existent. Thus, information is stored in the links, not in the nodes. The eventual addition of lower-level detail to the episodes will require the application of yet unknown qualitative, as opposed to quantitative, inference metrics.

Additionally, there is the issue of memory organization. Whenever researchers assume that specific concepts are organized in specific ways in human memory (i.e., "this MOP is connected to that MOP by this relationship"), it is nothing more than an educated guess. Currently, ATLAST's metrics depend more on the specific organization of memory, rather than on the content of memory, for correct operation. If the memory had been organized differently, so that there were a different number of links between certain MOPs, for example, ATLAST's interpretation of the input text would have been different. This particular realization of the metrics is not necessarily inaccurate, nor does the metrics' reliance on a particular organization of memory invalidate ATLAST, any more than similar educated guesses invalidate any other models of human understanding. This issue does remind us, however, that our implementation decisions can have as great an impact as our scientific theories on the perceived accuracy of our cognitive models, and that we should remain aware of where theoretical issues end and implementation issues begin.

Obviously, much work remains to be done on ATLAST. The current implementation has been applied only to short texts. In the future, we will process longer texts, and different types of texts, in order to discover additional rules for inference processing. The ATLAST model provides a framework for testing theories, as well as for making predictions which can be verified experimentally.

5.0 Conclusion

5.1 Summary

To some extent, ATLAST is a unification and refinement of ideas from previous models of human inference processes at the lexical, syntactic, and pragmatic levels. Yet, while ATLAST shares common features with each of these models, in many ways it is different from each of these same models. The features which distinguish the ATLAST model from others are discussed in greater detail in [Granger, Eiselt, & Holbrook, 1985]. A brief summary of those features follows:

- ATLAST unifies inference processing at three distinct levels: the lexical, syntactic, and pragmatic levels.
- The separation of intra-phrasal and inter-phrasal syntactic analysis enables ATLAST to process texts which humans understand and to make the same mistakes a human understander makes.
- The use of a spreading-activation memory model allows ATLAST to pursue competing inference paths simultaneously until syntactic or semantic information suggests otherwise. Previous models of inference decision processes either left a loose end or chose a default inference when faced with an ambiguity [DeJong, 1979; Granger, 1980; Lebowitz, 1980; Granger, 1981; Dyer, 1982; Wilensky, 1983].
- The concurrent operation of ATLAST's Capsulizcr, Proposer, and Filter permits pragmatic interpretations to be evaluated independently of syntactic decisions. This parallel organization also allows immediate evaluation and inhibition of competing inference paths, thus minimizing combinatorial explosion effects.
- ATLAST conforms to the results of controlled experiments on human subjects.

5.2 Final Comment

This paper describes how ATLAST attempts to understand only a five-word sentence. At first glance, this hardly seems like progress when one considers, for instance, that earlier systems understood hundreds of newspaper stories; in fact, it might even appear that work in natural language understanding is going backwards, at least from a performance perspective. What is really indicated by this phenomenon, though, is that we are becoming more aware of the great quantity of knowledge and the complexity of the processes which language understanders, both human and otherwise, must bring to bear in understanding even the simplest text. In this light, we should not measure the validity of any model of understanding in terms of how many stories it understands, how many words are in its vocabulary, or how fast it runs. More appropriately, we should ask such questions as: Is the model extensible? Does it compare favorably with experimental data? Is it learnable? Does it make testable predictions? In other words, cognitive models should be evaluated on the robustness of the theory which they embody. Only when that metric is satisfied will the engineering issues become relevant. From this perspective, it is safe to say that ATLAST is a step in the right direction.

6.0 Acknowledgments

Special thanks to Rick Granger and Jen Holbrook, whose efforts have made this work possible.

7.0 References

Anderson, J.R. *The Architecture of Cognition.* Cambridge, MA: Harvard University Press, 1983.

Charniak, E. Passing markers: A theory of contextual influence in language comprehension. *Cognitive Science,* 7, 171-190, 1983.

Cullingford, RE. *Script Application: Computer Understanding of Newspaper Stories.* Ph.D. thesis. Research Report #116. Department of Computer Science. Yale University, New Haven, CT, 1978.

DeJong, G.F. *Skimming Stories in Real Time: An Experiment in Integrated Understanding.* Ph.D. thesis. Research Report #158. Department of Computer Science. Yale University, New Haven, CT, 1979.

Dyer, M.G. *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension.* Cambridge, MA: MIT Press, 1983.

Fahlman, S.E. *NETL: A System for Representing and Using Real-World Knowledge.* Cambridge, MA: MIT Press, 1979.

Frazier, L., & Fodor, J.D. The sausage machine: A new two-stage parsing model. *Cognition,* 6, 291-325, 1978.

Granger, R.H When expectation fails: Towards a self-correcting inference system. *Proceedings of the First Annual National Conference on Artificial Intelligence,* Stanford, CA, 1980.

Granger, R.H. Directing and re-directing inference pursuit: Extra-textual influences on text interpretation. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence,* Vancouver, B.C., Canada, 1981.

Granger, R.H., Eiselt, K.P., & Holbrook, J.K. STRATEGIST: A program that models strategy-driven and content-driven inference behavior. *Proceedings of the National Conference on Artificial Intelligence,* Washington, D.C., 1983.

Granger, R.H., Eiselt, K.P., & Holbrook, J.K. Parsing with parallelism: A spreading-activation model of inference processing during text understanding. In J.L. Kolodner and C.K. Riesbeck (eds.), *Memory, Experience and Reasoning.* Hillsdale, NJ: Erlbaum, 1985 (to appear).

Granger, R.H., & Holbrook, J.K. Perseverers, recencies, and deferrers: New experimental evidence for multiple inference strategies in understanding. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society,* Rochester, NY, 1983.

Granger, R.H., Holbrook, J.K., & Eiselt, K.P Interaction effects between word-level and text-level inferences: Online processing of ambiguous words in context. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society,* Boulder, CO, 1984.

Kolodner, J.L. *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model.* Hillsdale, NJ: Erlbaum, 1984.

Lebowitz, M. *Generalization and Memory in an Integrated Vnderstanding System.* Ph.D. thesis. Research Report #186. Department of Computer Science. Yale University, New Haven, CT, 1980.

Lucas, M. Lexical access during sentence comprehension: Frequency and context effects. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society,* Rochester, NY, 1983.

Norvig, P. Six problems for story understanders. *Proceedings of the National Conference on Artificial Intelligence,* Washington, D C , 1983.

Quillian, MR. Semantic memory. In M. Minsky (ed.), *Semantic Information Processing.* Cambridge, MA: MIT Press, 1968.

Riesbeck, C.K., & Martin, C.E. *Direct Memory Access Parsing.* Research Report #354. Department of Computer Science. Yale University, New Haven, CT, 1985.

Schank, R.C. *Conceptual Information Processing.* Amsterdam: North-Holland, 1975.

Schank, R.C. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* New York: Cambridge University Press, 1982.

Seidenberg, M.S., Tanenhaus, M.K., Leiman, J.M., & Bienkowski, M. Automatic access of the meanings of ambiguous words in context: Some limitations of knowledge-based processing. *Cognitive Psychology,* 14, 489-537, 1982.

Small, S., Cottrell, G., & Shastri, L. Toward connectionist parsing. *Proceedings of the National Conference on Artificial Intelligence,* Pittsburgh, PA, 1982.

Swinney, D.A., & Hakes, D.T. Effects of prior context upon lexical access during sentence comprehension. *Journal of Verbal Learning and Verbal Behavior,* 15, 681-689, 1976.

Tanenhaus, M., Leiman, J., & Seidenberg, M. Evidence for multiple stages in processing of ambiguous words in syntactic contexts. *Journal of Verbal Learning and Verbal Behavior,* 18, 427-440, 1979.

Wilensky, R. *Understanding Goal-Based Stories.* Ph.D. thesis. Research Report #140. Department of Computer Science. Yale University, New Haven, CT, 1978.

Wilensky, R. *Planning and Understanding.* Reading, MA: Addison-Wesley, 1983.

Wilensky, R. Knowledge representation A critique and a proposal. *Proceedings of the First Annual Workshop on Theoretical Issues in Conceptual Information Processing,* Atlanta, GA, 1984.