

INTEGRATING TOP-DOWN AND BOTTOM-UP STRATEGIES IN A TEXT PROCESSING SYSTEM

Lisa F. Rau and Paul S. Jacobs
Artificial Intelligence Branch
GE Company, Corporate R&D
Schenectady, NY 12301

Abstract

The SCISOR system is a computer program designed to scan naturally occurring texts in constrained domains, extract information, and answer questions about that information. The system currently reads newspaper stories in the domain of corporate mergers and acquisitions. The language analysis strategy used by SCISOR combines full syntactic (bottom-up) parsing and conceptual expectation-driven (top-down) parsing. Four knowledge sources, including syntactic and semantic information and domain knowledge, interact in a flexible manner. This integration produces a more robust semantic analyzer designed to deal gracefully with gaps in lexical and syntactic knowledge, transports easily to new domains, and facilitates the extraction of information from texts.

INTRODUCTION

The System for Conceptual Information Summarization, Organization and Retrieval (SCISOR) is an implemented system designed to extract information from naturally occurring texts in constrained domains. The derived information is stored in a conceptual knowledge base and retrieved using a natural language analyzer and generator.

Conceptual information extracted from texts has a number of advantages over other information-retrieval techniques [Rau, 1987a], in addition to allowing for the automatic generation of databases from texts.

The integration of top-down, expectation driven processing, and bottom-up, language-driven parsing is important for text understanding. *Bottom-up* strategies identify surface linguistic relations in the input and produce conceptual structures from these relations. With the input "ACE made ACME an offer", a good "bottom-up" linguistic analyzer can identify the subject, verb, direct and indirect objects. It also can determine that ACME was the recipient of an offer, rather than being made into an offer, as in "ACE made ACME a subsidiary".

Top-down methods use extensive knowledge of the context of the input, practical constraints, and conceptual expectations based on previous events to fit new information into an existing framework. A good "top-down" analyzer might determine from "ACE made ACME an offer" that ACME is the target of a takeover (which is not obvious

from the language, since the offer could be for something that ACME owns), and relate the offer to other events (previous rumors or competing offers).

Bottom-up methods tend to produce more accurate parses and semantic interpretations, account for subtleties in linguistic expression, and detect inconsistencies and lexical gaps. Top-down methods are more tolerant of unknown words or grammatical lapses, but are also more apt to derive erroneous interpretations, fail to detect inconsistencies between what is said and how it is interpreted, and often cannot produce any results when the text presents unusual or unexpected information. Integration of these two approaches can improve the depth and accuracy of the understanding process.

SCISOR is unique in its integration of the bottom-up processing performed by its analyzer, TRUMP (Transportable Understanding Mechanism Package) [Jacobs, 1986], with other sources of information in the form of conceptual expectations.

In this paper, four information sources are described that are used by SCISOR to produce meaning representations from texts. The actual processing sequence and timing of the application of these sources are illustrated.

THE SCISOR SYSTEM

The SCISOR system is currently being tested with newspaper stories about corporate takeovers. The domain provides interesting subject matter as well as some rich language. The gradual development of the stories over time motivates a natural language approach, while the restricted nature of the material allows us to encode conceptual expectations necessary for top-down processing.

The following is an example of the operation of SCISOR on a simple news story:

W ACQUISITION UPS BID FOR WARNACO
Warnaco received another merger offer, valued at \$36 a share, or \$360 million. The buyout offer for the apparel maker was made by the W Acquisition Corporation of Delaware.

User: *Who took over Warnaco?*

System: W Acquisition offered \$36 per share for Warnaco.

User: *What happened to Warnaco last Tuesday?*

System: Warnaco rose 2 1/2 as a result of rumors.

The system has been demonstrated with a small set of input texts, and is being expanded to handle large numbers of newswire stories using a large domain knowledge base and substantial vocabulary.

SOURCES OF INFORMATION

Text processing in SCISOR is accomplished through the integration of sources of knowledge and types of processing. The four sources of knowledge that SCISOR uses to extract meaning from text are as follows:

- A. Role-filler Expectations:** Constraints on what can fill a conceptual role are the primary source of information used in top-down processing.
- B. Event Expectations:** Expectations about events that may occur in the future are created from previous stories, and used to predict values in the expected events if they occur.
- C. Linguistic:** Grammatical, lexical and phrasal knowledge is used whenever it is available and reliable. Sub-language (domain-specific) linguistic information may also be used if available.
- D. World Knowledge Expectations:** World knowledge expectations can disambiguate multiple interpretations through domain-specific heuristics.

SCISOR can operate with any combination of these information sources. When one or more sources are lacking, the information extracted from the texts may be more superficial, or less reliable. The flexibility in depth of processing provided by these multiple information sources is an interesting feature in its own right, in addition to forming the foundations for a system to "skim" efficiently when a new text contains material already processed.

As an example of each source of information, consider the following segment from the text printed previously:

Warnaco received another merger offer,
valued at \$36 a share, or \$360 million.

Role-filler expectations allow SCISOR to make reliable interpretations of the dollar figures in spite of incomplete lexical knowledge of the syntactic roles they play in the sentence. This is accomplished because prices of stock are constrained to be "small" numbers, whereas fillers of takeover-bid value roles are constrained to be "large" quantities. Event expectations lead to the deeper interpretation that this offer is an increase over a previous offer because one expects some kind of rebuttal to an offer to occur in the future. An increased offer is one such rebuttal. World knowledge might allow the system to predict whether the offer was an increase or a competing offer, depending on what other information was available.

A unique feature of SCISOR is that partial linguistic knowledge contributes to all of these interpretations, and to the understanding of "received" in this context. This is noteworthy because general knowledge about "receive" in

this case interacts with domain knowledge in understanding the role of Warnaco in the offer.

A robust parser and semantic interpreter could obtain these features from the texts without the use of expectations. This would make top-down processing unnecessary. Robust in-depth analysis of texts, however, is beyond the near-term capabilities of natural language technology; thus SCISOR is designed with the understanding that there will always be gaps in the system's knowledge base that must be dealt with gracefully.

Now the four sources of information used to extract information are described in more detail, followed by a discussion of how they interact in the processing of two sample texts.

A. Role-filler Expectations

The simplest kind of expectation-driven information that can be used is termed "role-filler" expectations. These expectations take the form of constraints on the filler of a conceptual role. This is the primary source of processing power in expectation-driven systems such as FRUMP [De-Jong, 1979]. The following list illustrates some examples of constraints on certain fillers of roles used in the corporate takeover domain.

ROLE	FILLER-CONSTRAINT	EXAMPLE
target	company-agent	ACE
suitor	company-agent	ACME
price-per-share	small number	\$45
total value	large number	\$45 million

This information is encoded declaratively in the knowledge base of the system. During the processing of a text, roles may be filled with more than one hypothesis; however, as soon as a filler for a role is certain, the process of elimination is used to aid in the resolution of other bindings. Thus, if SCISOR determines that ACE is a takeover target, it will assume by default that ACME is the suitor if the two companies appear in the same story and there is no additional information to aid in the disambiguation.

B. Event Expectations

Expectations that certain events will occur in the future are a second source of information available to aid in the interpretation of new events. These expectations arise from the events in previous stories. For example, when the system reads that rumors have been swirling around ACE as a takeover target, an event expectation is set up that anticipates an offer for ACE in some future story. When an offer has been made, expectations are set up that some kind of rebuttal will take place. This rebuttal may be a rejection or an acceptance of the offer. The acceptance of the offer option carries with it the event expectation that the total value of the takeover will be the amount of the offer.

Event expectations are implemented as domain-dependent, declarative properties of the events in the do-

main. They are derived from the script-like [Schank and Abelson, 1977] representations of typical event sequences.

C. Linguistic Analysis

The most important source of information used in text processing is a full bottom-up parser. TRUMP is a flexible language analyzer consisting of a syntactic processor and semantic interpreter [Jacobs, 1986, Jacobs, 1987a]. The system is designed to fill conceptual roles using linguistic, conceptual, and metaphorical relationships distributed in a knowledge hierarchy.

Within SCISOR, TRUMP identifies linguistic relationships in the input, using lexical and syntactic knowledge. Knowledge structures produced by TRUMP are passed through an interface to the top-down processing components. Pieces of knowledge structures may then be tied together with the expectation-driven processing components.

In the case of a complete parse of an input sentence, the knowledge structures produced by TRUMP contain most of the structure of the final interpretation, although expectations often further refine the analysis. In the case of partial parses, more of the structure is determined by role-filler expectations. The following are two simple examples of this division of labor:

Input:

W Acquisition offered \$36 a share for Warnaco.

Partial parser output:

```
(Basic-sentence
  (NP (Name-NP
      (Name (C-Name W_Acquisition))))
  (VP (Adjunct-VP
      (VP (Transitive-VP
          (Verb-part (Basic-verb-part
                    (V offered)))
          (NP (Postmodified-NP
              (NP ($-NP $36))
              (MOD (Ratio-modifier
                  (R-word a) (N share))))))
          (PP (Basic-PP (P for) (NP (Name-NP
              (G-Name Warnaco)))))))
```

TRUMP interpretation:

```
(offer
  (offerer W-Acq-Co)
  (offeree Warnaco)
  (offer (dollars (quantity 36)
            (denominator share))))
```

Final interpretation:

```
(corp-takeover-offer
  (suitor W-Acq-Co)
  (target Warnaco)
  (dps (quantity 36)))
```

Input:

Warnaco received another merger offer, valued at \$36 a share.

Partial parser output:

```
(Subj-verb-relation
  (Subj (NP (Name-NP (Name
                    (C-Name W_Acquisition))))
  (Verb (V received)))
(N offer)
(NP (Postmodified-NP (NP ($-NP $36))
  (MOD (Ratio-modifier (R-word (det a))
                    (Noun-part (N share))))))
```

TRUMP interpretation:

```
(offer
  (transfer-event (recipient Warnaco))
  (dollars (quantity 36) (denominator share)))
```

Final interpretation:

```
(corp-takeover-offer
  (target Warnaco)
  (dps (quantity 36)))
```

In the first example above, TRUMP succeeds in producing a complete syntactic parse, along with the corresponding semantic interpretation. The domain knowledge helps only to specify the verb sense of "offer". In the second example, however, more of the work is done by the domain-dependent expectations. In this case, the unknown words prevent TRUMP from completing the parse, so the output from the parser is a set of linguistic relations. These relations allow the semantic interpreter to produce some superficial conceptual structures, but the final conceptual roles are filled using domain knowledge.

The distinction between the general *offer* and the more specific *corp-takeover-offer* is essential for understanding texts of this type. In general, an offer may be made *for* something *to* someone, but it is only in the corporate takeover domain that the target of the takeover (the *for* role) is by default the same as the recipient of the offer (the *to* role). Since TRUMP is a domain-independent analyzer, it cannot itself fill such roles appropriately.

The knowledge sources at work in SCISOR and the timing of the information exchange in the system are described in the next section.

D. World Knowledge Expectations

If all the above sources of information are still insufficient to determine or satisfactorily disambiguate potential relationships between items in the text, so called "world knowledge" can be called into play. This world knowledge takes the form of domain-dependent generalizations, implemented as declarative relationships between concepts. For example, in the corporate takeover domain, a piece of world knowledge that can aid in the determination of what company is taking over what company is the following:

If it is ambiguous whether:
Company A is taking over Company B
or Company B is taking over Company A

Choose the larger company to be the suitor
and the smaller company to be the target

This example uses the knowledge that it is almost always the case that the suitor is larger than the target company. The utilization of this generalization (that typically larger companies take over smaller companies) requires knowledge of the company sizes, assumed to be present in the knowledge base of the system. Another example is:

If it is ambiguous whether:
value A is a previous offer or present
stock price and value B is a new offer
or vice versa,

Choose the larger offer for the new offer
or present stock price, and the smaller
offer for the previous offer

In this case, a company rarely would *decrease* their offer unless something unexpected happened to the target company before the takeover was completed. Similarly, an offer is almost always for more than the current value of the stock on the stock market.

These heuristics incorporate expectations that arise from potentially complex explanations. For example, the reason *why* a new offer is higher than an old offer rests on a complex understanding of the plan of the suitor to reach their goal of taking over the target company. The world knowledge presented here represents a compilation of this complex reasoning into simple heuristics for text understanding, albeit ad hoc.

Although this type of information is shown in a rule-like form, it is implemented with special relationship links that contain information as to how to compute the truth value of the relationship. When this type of knowledge is needed to disambiguate an input, the system checks if any objects have these "world knowledge constraints". If so, they are activated and applied to the situation under consideration.

The intuition underlying the inclusion of heuristics of this sort is that there is a great deal of "common sense" information that can increase an understanding mechanism's ability to extract meaning. This type of information is a last resort for determining conceptual relations when other more principled sources of information are exhausted.

KNOWLEDGE INTEGRATION

Each of the four sources of information described above is utilized at different points in the processing of the input text, and with different degrees of confidence. The following algorithm describes a particular instantiation of this order for a hypothetical event sequence involving rumors about ACE, followed by an offer by ACME for ACE.

In general, event expectations are set up as soon as an event that has an expectation property is detected. That is, as soon as the system sees a rumor, it sets up an expectation that there will be an offer for the company the rumor was about sometime in the future.

When that event-expectation is confirmed, those expectations are realized and the information expected is added to the meaning extracted from the text being processed. Note that these realized expectations may later be retracted given additional information. Role-filler expectations then create multiple hypotheses about which items may fill what roles. These are narrowed down by any constraints already present by event expectations.

Linguistic analysis, when it provides a complete final meaning representation for a portion of the text containing features of interest, always supercedes a conflicting role-filler expectation. For example, if a role-filler expectation hypothesized that ACE was the target in a takeover, and the parser determined that ACME was the object of the takeover, ACME alone would be included as the target.

World knowledge expectations are invoked only in the case of conflicting or ambiguous interpretations. For example, if after all the processing is finished and the system does not know whether ACE is taking over ACME or vice versa, the expectation that the larger company is typically the suitor is invoked and used in the final disambiguation.

Below are the sample input texts, followed by the sequence of steps that are taken by the program.

ACE, an apparel maker planning a leveraged buyout, rose \$2 1/2 to \$35 3/8, as a rumor spread that another buyer might appear. The company said there were no corporate developments to account for the rise, and the rumor could not be confirmed.

----- later on -----
ACE received another merger offer, valued at \$36 a share, or \$360 million. The buyout offer for the apparel maker was made by the ACME Corporation of Delaware. ACE closed yesterday at \$35 3/8.

1. System reads first story and extracts information that there are rumors about ACE and that the stock price is currently \$35 3/8, using role-filler expectations.
2. An event expectation is set up that there will be an offer-event, with ACE as the target of the takeover offer.
3. System begins reading story involving a takeover offer and ACE.
4. Target slot of offer is filled with ACE from the event expectation.
5. An event expectation is set up that there will be a rebuttal to the offer sometime in the future.
6. System encounters ACME which it knows to be a company. Suitor slot of offer is thus filled with ACME via a role-filler expectation.

7. *\$36 a share* is parsed with the phrasal lexicon.
8. *\$36 a share* is added as a candidate for either the stock's current price on the stock market or the amount of the ACME offer, due to role-filler expectations.
9. *\$360 million* is parsed with the phrasal lexicon.
10. *\$360 million* is added as candidate for the total value of the offer due to a role-filler expectation that expects total values to be large numbers.
11. Syntactic and semantic analysis determine that the offerer is ACME, and the target is ACE. This reinforces the interpretations previously hypothesized.
12. Syntactic and semantic analysis determine the location of the ACME Corporation to be Delaware.
13. *\$35 3/8* is encountered, which is taken to be a price-per-share amount, due to a role-filler expectation that expects prices per share to be small numbers.
14. *\$35 3/8 a share* is added as a candidate for either the stock's current price on the stock market or the amount of the ACME offer.
15. *\$35 3/8* is taken to be the stock's current price and *\$36* is taken to be the amount of the ACME offer, due to the world knowledge expectation that expects the offer to exceed the current trading price.

The contribution of the various sources of knowledges varies with the amount of knowledge they can be brought to bear on the language being analyzed. That is, given more syntactic and semantic knowledge, TRUMP could have done more work in the analyses of these stories. Given more detailed conceptual expectations, the bottom-up mechanism also could have extracted more meaning. Together, the two mechanisms should combine to produce a deeper and more complete meaning representation than either one could alone.

IMPLEMENTATION

SCISOR consists of a variety of programs and tools, operating in conjunction with a declarative knowledge base of domain-independent linguistic, grammatical and world knowledge and domain-dependent lexicon and domain knowledge. A brief overview of the system may be found in [Rau, 1987c], and a more complete description in [Rau, 1987b]. The natural language input is processed with the TRUMP parser and semantic interpreter [Jacobs, 1986]. Linguistic knowledge is represented using the Ace linguistic knowledge representation framework [Jacobs and Rau, 1985]. Answers to user's questions and event expectations are retrieved using the retrieval mechanism described in [Rau, 1987b]. Responses to the user will be generated with the KING [Jacobs, 1987b] natural language generator when that component is integrated with SCISOR; currently output is "canned". The events in SCISOR are represented using the KODIAK knowledge representation

language [Wilensky, 1986], augmented with some scriptal knowledge of typical events in the domain.

SYSTEM STATUS

All the components of SCISOR described here have been implemented, although not all have been connected together. The system can, as of this writing, process a number of stories in the domain. The processing entails the combined expectation-driven and language driven capabilities described here. For questions that the system can understand, SCISOR retrieves conceptual answers to input questions. These answers are currently output using pseudo-natural language, but we are in the process of integrating the KING generator.

SCISOR is currently being connected to an automatic source of on-line information (a newswire) for extensive testing and experimentation. The goal of this effort is to prove the utility of the system for processing large bodies of text in a limited domain.

Although there will undoubtedly be many lessons in extending SCISOR to handle thousands of texts, SCISOR's first few stories have already demonstrated some of the advantages of the approach described here:

1. Much of the knowledge used in analyzing these stories is domain-independent.
2. Where top-down strategies fail, SCISOR can still extract *some* information from the texts and use this information in answering questions.
3. Unknown words (lexical gaps) and grammatical lapses are tolerated.

These three characteristics simply cannot be achieved without combining top-down and bottom-up strategies.

The major barrier to the practical success of text processing systems like SCISOR is the vast amount of knowledge required to perform accurate analysis of any body of text. This bottleneck has been partially overcome by the graceful integration of processing strategies in the system; the program currently operates using only hundreds of known words. However, SCISOR is designed to benefit ultimately from an extended vocabulary (i. e. thousands of word roots) and increased domain knowledge. The vocabulary and knowledge base of the system are constantly being extended using a combination of manual and automated techniques.

EXTENSIBILITY AND PORTABILITY

Our research has combined some of the advantages of top-down language processing methods (tolerance of unknown inputs, understanding in context) with the assets of bottom-up strategies (broader linguistic capabilities, partial results in the absence of expectations). The system described here competently answers questions about constrained texts, uses the same language analyzer for text processing and question answering, and has been applied to other domains as well as the corporate takeover stories. SCISOR is thus a state-of-the-art system, but like

other text processing systems the main chore that remains is to allow for the practical extraction of information from thousands of real texts. The following are the main issues involved in making such a system a reality and how we address them:

Lexicon design: The size of the text-processing lexicon is important, but sheer vocabulary is not of much help. What is needed is a lexicon that accounts both for the basic meanings of common words and the specialized use of terms in a given context. We use a *hierarchical phrasal lexicon* [Besemer and Jacobs, 1987, Dyer and Zernik, 1986] to allow domain-specific vocabulary to take advantage of existing linguistic knowledge and ultimately to facilitate automatic language acquisition.

Grammar: A disadvantage of many approaches to text processing is that it is counterintuitive to assume that most language processing is domain-specific. While specialized knowledge is essential, a portable grammar, like a core lexicon, is indispensable. Language is too complex to be reduced to a few domain-specific heuristics. Because specialized constructs may inherit from general grammatical rules, TRUMP allows specialized sublanguage grammar to interact with "core" grammar. It is still a challenge, however, to deal gracefully with constructs in a sublanguage that would ordinarily be extragrammatical.

Conceptual Knowledge: The KODIAK knowledge representation, used for conceptual knowledge in SCISOR, allows for multiple inheritance as well as structured relationships among conceptual roles. This representation is useful for the retrieval of conceptual information in the system. A broader base of "common sense" knowledge in KODIAK will be used to increase the robustness of SCISOR.

Our strategy has been to attack the robustness problem by starting with the underlying knowledge representation issues. There will be no way to avoid the work involved in scaling up a system, but with this strategy we hope that much of this work will be useful for text processing in general, as well as for analysis within a specialized domain.

FUTURE DIRECTIONS

In the immediate future, we hope to connect SCISOR to a continuous source of on-line information to begin collecting large amounts of conceptually analyzed material, and extensively testing the system.

We also plan to dramatically increase the size of the lexicon through the addition of an on-line source of dictionary and thesaurus information. The system grammar also will increase in coverage over time, as we extend and improve the capabilities of the bottom-up TRUMP parser.

Another interesting extension is the full implementation of a parser skimming mode. This mode of operation, triggered when the system recognizes input events that are identical to events it has already read about, will cause the

parser to perform very superficial processing of the text. This superficial or skimming processing will continue until the parser reaches a point in the text where the story is no longer reporting on events the system has already read about.

RELATED RESEARCH

The bulk of the research on natural language text processing adheres to one of the two approaches integrated in SCISOR. The practical issue for text processing systems is that it is still far from feasible to design a program that processes extended, unconstrained text. Within the "bottom-up" framework, one of the most successful strategies, in light of this issue, is to define a specialized domain "sublanguage" [Kittredge, 1982] that allows robust processing so long as the texts use prescribed vocabulary and linguistic structure. The "top-down" approach similarly relies heavily on the constraints of the textual domain, but in this approach the understanding process is bound by constraints on the knowledge to be derived rather than restrictions on the linguistic structures.

The bottom-up, or language-driven strategy, has the advantage of covering a broad class of linguistic phenomena and processing even the more intricate details of a text. Many systems [Grishman and Kittredge, 1986] have depended on this strategy for processing messages in constrained domains. Other language-driven programs [Hobbs, 1986] do not explicitly define a sublanguage but rely on a robust syntax and semantics to understand the constrained texts. These systems build upon existing grammars, which may make the semantic interpretation of the texts difficult.

The top-down, or expectation-driven, approach, offers the benefit of being able to "skim" texts for particular pieces of information, passing gracefully over unknown words or constructs and ignoring some of the complexities of the language. A typical, although early, effort at skimming news stories was implemented in FRUMP [DeJong, 1979], which accurately extracted certain conceptual information from texts in preselected topic areas. FRUMP proved that the expectation-driven strategy was useful for scanning texts in constrained domains. This strategy includes the banking telex readers TESS [Young and Hayes, 1985] and ATRANS [Lytinen and Gershman, 1986]. These programs all can be easily "fooled" by unusual texts, and can obtain only the expected information.

The difficulty of building a flexible understanding system inhibits the integration of the two strategies, although some of those mentioned above have research efforts directed at integration. Dyer's BORIS system [Dyer, 1983], a program designed for in-depth analysis of narratives rather than expository text scanning, integrates multiple knowledge sources and, like SCISOR, does some dynamic combination of top-down and bottom-up strategies. The linguistic knowledge used by BORIS is quite different from that of TRUMP, however. It lacks explicit syntactic struc-

tures and thus, like the sublanguage approach, relies more heavily on domain-specific linguistic knowledge. Lytinen's MOPTRANS [Lytinen, 1986] integrates syntax and semantics in understanding, but the syntactic coverage of the system is in no way comparable to the bottom-up programs. SCISOR is, to our knowledge, the first text processing system to integrate full language-driven processing with conceptual expectations.

CONCLUSION

The analysis of extended texts presents an extremely difficult problem for artificial intelligence systems. Bottom-up processing, or linguistic analysis, is necessary to avoid missing information that may be explicitly, although subtly, conveyed by the text. Top-down, or expectation-driven processing, is essential for the understanding of language in context. Most text analysis systems have relied too heavily on one strategy.

SCISOR represents a unique integration of knowledge sources to achieve robust and reliable extraction of information from naturally occurring texts in constrained domains. Its ability to use lexical and syntactic knowledge when available separates it from purely expectation-driven semantic analyzers. At the same time, its lack of reliance on any single source of information and multiple "fall-back" heuristics give the system the ability to focus attention and processing on those items of particular interest to be extracted.

REFERENCES

- [Besemer and Jacobs, 1987] David Besemer and Paul S. Jacobs. FLUSH: a flexible lexicon design. In *Proceedings of the 25th Meeting of the Association for Computational Linguistics*, Palo Alto, California, 1987.
- [DeJong, 1979] Gerald DeJong. *Skimming Stories in Real Time: An Experiment in Integrated Understanding*. Research Report 158, Department of Computer Science, Yale University, 1979.
- [Dyer, 1983] Michael G. Dyer. *In-Depth Understanding*. MIT Press, Cambridge, MA, 1983.
- [Dyer and Zernik, 1986] Michael G. Dyer and Uri Zernik. Encoding and acquiring meanings for figurative phrases. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, New York, 1986.
- [Grishman and Kittredge, 1986] Ralph Grishman and Richard Kittredge, editors. *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*. Lawrence Erlbaum, Hillsdale, NJ, 1986.
- [Hobbs, 1986] Jerry R. Hobbs. Site report: overview of the TACITUS project. *Computational Linguistics*, 12(3):220-222, 1986.
- [Jacobs, 1986] Paul S. Jacobs. Language analysis in not-so-limited domains. In *Proceedings of the Fall Joint Computer Conference*, Dallas, Texas, 1986.
- [Jacobs, 1987a] Paul S. Jacobs. A knowledge framework for natural language analysis. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, 1987.
- [Jacobs, 1987b] Paul S. Jacobs. Knowledge-intensive natural language generation. *Artificial Intelligence*, 33(3):325-378, November 1987.
- [Jacobs and Rau, 1985] Paul S. Jacobs and Lisa F. Rau. Ace: associating language with meaning. In Tim O'Shea, editor, *Advances in Artificial Intelligence*, pages 295-304, North Holland, Amsterdam, 1985.
- [Kittredge, 1982] Richard Kittredge. Variation and homogeneity of sublanguages. In Richard Kittredge and John Lehrberger, editors, *Sublanguages: Studies of Language in Restricted Domains*, pages 107-137, Walter DeGruyter, New York, 1982.
- [Lytinen, 1986] Steven Lytinen. Dynamically combining syntax and semantics in natural language processing. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, 1986.
- [Lytinen and Gershman, 1986] Steven Lytinen and Anatole Gershman. ATRANS: automatic processing of money transfer messages. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, 1986.
- [Rau, 1987a] Lisa F. Rau. Information retrieval in never-ending stories. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 317-321, Los Altos, CA, Morgan Kaufmann Inc., Seattle, Washington, July 1987.
- [Rau, 1987b] Lisa F. Rau. Knowledge organization and access in a conceptual information system. *Information Processing and Management, Special Issue on Artificial Intelligence for Information Retrieval*, 23(4):269-283, 1987.
- [Rau, 1987c] Lisa F. Rau. SCISOR: a system for effective information retrieval from text. In *Poster Session Proceedings of the Third IEEE Conference on Artificial Intelligence Applications*, Orlando, FL, 1987.
- [Schank and Abelson, 1977] Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum, Halsted, NJ, 1977.
- [Wilensky, 1986] R. Wilensky. Knowledge representation - a critique and a proposal. In Janet Kolodner and Chris Riesbeck, editors, *Experience, Memory, and Reasoning*, pages 15-28, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986.
- [Young and Hayes, 1985] S. Young and P. Hayes. Automatic classification and summarization of banking telexes. In *The Second Conference on Artificial Intelligence Applications*, pages 402-208, IEEE Press, 1985.