# Annotating the World Wide Web using Natural Language

**Boris Katz**
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
545 Technology Square
Cambridge, MA 02139, USA
boris@ai.mit.edu

This paper describes the START Information Server built at the MIT Artificial Intelligence Laboratory. Available on the World Wide Web since December 1993, the START Server provides users with access to multi-media information in response to questions formulated in English. Over the last 3 years, the START Server answered hundreds of thousands of questions from users all over the world.

The START Server is built on two foundations: the sentence-level Natural Language processing capability provided by the START Natural Language system (Katz [1990]) and the idea of natural language annotations for multi-media information segments. This paper starts with an overview of sentence-level processing in the START system and then explains how annotating information segments with collections of English sentences makes it possible to use the power of sentence-level natural language processing in the service of multi-media information access. The paper ends with a proposal to annotate the World Wide Web.

## An Overview of the START system

The START natural language system (SynTactic Analysis using Reversible Transformations) consists of two modules which share the same grammar (Katz [1980]). The *understanding* module analyzes English text and produces a *knowledge base* which incorporates the information found in the text. Given an appropriate segment of the knowledge base, the *generating* module produces English sentences. A user can retrieve the information stored in the knowledge base by querying it in English. The system will then produce an English response.

START has been used by researchers at MIT and other universities and research laboratories for constructing and querying knowledge bases using English. (Katz and Winston [1983], Winston *et al* [1983], Doyle [1984], Katz and Brooks [1987], Keshi and Katz [1991], Winston [1992], Katz [1994]).[1]

[1] For other approaches to the design of natural language querying systems, see, for example, Warren and Pereira [1982], Shapiro and Rapaport [1987], Allen and Schubert [1991], and others.

Given an English sentence containing various relative clauses, appositions, multiple levels of embedding, *etc*, the START system first breaks it up into smaller units, called *kernel* sentences (usually containing one verb). After separately analyzing each kernel sentence, START rearranges the elements of all parse trees it constructs into a set of embedded representational structures. These structures are made up of a number of fields corresponding to various syntactic parameters of a sentence, but the three most salient parameters, the subject of a sentence, the object, and the relation between them are singled out as playing a special role in indexing. These parameters are explicitly represented in a discrimination network for efficient retrieval. As a result, all sentences analyzed by START are indexed as embedded *ternary expressions (T-expressions)*, **<subject relation object>**. Certain other parameters (adjectives, possessive nouns, prepositional phrases, *etc.*) are used to create additional T-expressions in which prepositions and several special words may serve as relations. For instance, the following simple sentence

(1) Bill surprised Hillary with his answer

will produce two T-expressions:

(2) **<<Bill surprise Hillary> with answer>**
**<answer related-to Bill>**

The remaining parameters—adverbs and their position, tense, auxiliaries, voice, negation, *etc.*—are recorded in a representational structure called a *history*. The history has a *page* pertaining to each sentence which yields the given T-expression. When we index the T-expression in the knowledge base, we cross-reference its three components and attach the history to it. One can thus think of the resulting entry in the knowledge base as a "digested summary" of the syntactic structure of an English sentence.

In order to handle embedded sentences, START allows any T-expression to take another T-expression as its subject or object. START can analyze and generate sentences with arbitrarily complex embedded structures.

Questions are requests for information from START's knowledge base. In order to answer a question START must translate the question into a T-expression template which can be used to search the knowledge base for T-expressions which contain information relevant to providing an answer to the question. Let us assume that as a result of analyzing and indexing a text containing sentence (1), the knowledge base currently includes T-expressions (2). Now suppose that a user asks START the following *wh*-question:

(3) Whom did Bill surprise with his answer?

In the context of (1), the answer is *Hillary*. In order to determine this, the system must first turn the question (3) into a T-expression template that can be used to search the knowledge base. The first step in this process is to undo the effects of the *wh*-movement transformation that is used to create English *wh*-questions. To do this, START must find the place in sentence (3) where the *wh*-word *whom* came from and then insert the *wh*-word in this position:

(4) Bill surprised *whom* with his answer.

Next the language understanding system leads sentence (4) through the same flow of control as any declarative sentence and produces the following T-expressions which serve as patterns used to query the knowledge base:

(5) **<<Bill surprise** *whom* **> with answer>**
**<answer related-to Bill>**

Treating *whom* as a matching variable, the system feeds query (5) through a matcher in order to determine whether there is anything in the knowledge base that matches (5). The matcher finds T-expressions (6) created from sentence (1):

(6) **<<Bill surprise Hillary> with answer>**
**<answer related-to Bill>**

and the language generation system then uses these T-expressions to produce the English response to question (3):

(7) Bill surprised Hillary with his answer.

START handles *yes-no* questions in a similar fashion. Suppose that START had been asked the *yes-no* question:

(8) Did Bill surprise Hillary with his answer?

As in the *wh*-case, START would turn this question into a T-expression template that could be matched against the T-expressions in the knowledge base. The difference between *yes-no* and *wh*-questions is that the T-expression templates generated by a *yes-no* question would contain no *wh*-variables. Still, the match will be found allowing the system to answer:

(9) Yes, Bill surprised Hillary with his answer.

## Introducing S-rules

The T-expressions in the START system are built using the pattern **<subject relation object>** at every level of embedding and thus mimic the hierarchical organization of English sentences and parallel the representational characteristics of natural language. A language-based knowledge representation system has many advantages: it is very expressive and easy to use; it provides a uniform symbolic representation for parsing and generation; and it makes it possible to automatically create large knowledge bases from natural language texts.

However, a representation mimicking the hierarchical organization of natural language syntax has one undesirable consequence: sentences differing in their surface syntax but close in meaning are not considered similar by the system. Thus, given sentence (10) as input, START will create T-expressions (11), whereas a near paraphrase, sentence (12), will generate T-expressions (13):

(10) Bill surprised Hillary with his answer.

(11) **<<Bill surprise Hillary> with answer>**
**<answer related-to Bill>**

(12) Bill's answer surprised Hillary.

(13) **<answer surprise Hillary>**
**<answer related-to Bill>**

Speakers of English know (at least implicitly) that in sentence (10), the subject (*Bill*) brings about the emotional reaction (*surprise*) by means of some property expressed in the *with* phrase. Sentence (12) describes the same emotional reaction as in (10) despite different syntactic realizations of some of the arguments; namely, in (12), the property and its possessor are collapsed into a single noun phrase. It seems natural that this kind of knowledge be available to a natural language system. However, START, as described so far, does not consider T-expressions (11) and (13), which are associated with these sentences, to be similar.

The difference in the T-expressions becomes particularly problematic when START is asked a question. Suppose the input text includes the *surprise* sentence (10) that is stored in the knowledge base using T-expressions (11). Now suppose the user asked the following question:

(14) Whose answer surprised Hillary?

Although a speaker of English could easily answer this question after being told sentence (10), START would not be able to answer it because T-expressions (15) produced by question (14) will not match T-expressions (11) in the knowledge base.

(15) **<answer surprise Hillary>**
**<answer related-to** *whom***>**

To be able to handle such questions, the START system should be made aware of the interactions between the syntactic and semantic properties of verbs. Interactions similar to the one just described pervade the English language and, therefore, cannot be ignored in the construction of a natural language system.

The *surprise* example illustrates that START needs information that allows it to deduce the relationship between alternate realizations of the arguments of verbs. In this instance, we want START to know that whenever *A surprised B with C*, then it is also true that *A's B surprised C*. We do this by introducing rules that make explicit the relationship between alternate realizations of the arguments of verbs. We call such rules *S-rules*. Here is the S-rule that solves the problem caused by the verb *surprise*:[2]

(16) *Surprise* S-rule
**If** <<subject **surprise** object1> **with** object2>
**Then** <object2 **surprise** object1>

S-rules are implemented as a rule-based system. Each S-rule is made up of two parts, an antecedent (the **If**-clause) and a consequent (the **Then**-clause). Each clause consists of a set of templates for T-expressions, where the template elements are filled by variables or constants. The *Surprise* S-rule will apply only to T-expressions which involve the verb *surprise* and which meet the additional structural constraints.

S-rules operate in two modes: *forward* and *backward*. When triggered by certain conditions, S-rules in the forward mode allow the system to intercept T-expressions produced by the understanding module, transform or augment them in a way specified by the rule, and then incorporate the result into the knowledge base. For instance, if the *Surprise* S-rule is used in the forward mode, as soon as its antecedent matches T-expressions (17) produced by the understanding module, it creates a new T-expression in (18) and then adds it to the knowledge base:

(17) **<<Bill surprise Hillary> with answer>**
**<answer related-to Bill>**

(18) **<answer surprise Hillary>**
**<answer related-to Bill>**

Now question (14) can be answered since T-expressions (15) associated with this question match against T-expressions (18). The generating module of START responds:

(19) Bill's answer surprised Hillary.

All additional facts produced by the forward S-rules are instantly entered in the knowledge base. The forward mode is especially useful when the information processed by START is put into action by another computer system because in such a situation START ought to provide the interfacing system with as much data as possible.

In contrast, the backward mode is employed when the user queries the knowledge base. Often for reasons of computational efficiency, it is advantageous not to incorporate all inferred knowledge into the knowledge base immediately. S-rules in the backward mode trigger when a request comes in which cannot be answered directly, initiating a search in the knowledge base to determine if the answer can be deduced from the available information. For example, the *Surprise* S-rule used in the backward mode does *not* trigger when sentence (10) is read and T-expressions (11) are produced by START; it triggers only when question (14) is asked.

## The Lexical Component of START

In order to understand an English sentence, the START system needs to have access to morphological, syntactic, and semantic information about the words in the sentence. All the words that the system is aware of, along with information about their part of speech, inflection, gender, number, *etc.* are stored in the *Lexicon*. Virtually every branch of START uses the Lexicon to accomplish its task. In this section we discuss the way in which the Lexicon extends the system's ability to deal with semantic-syntactic interdependencies. We show that the Lexicon provides a place where a verb's membership in a semantic class can be registered, allowing more general S-rules to be stated.

Note that formulating a special purpose S-rule which applies only to the verb *surprise* does not seem to be the best solution to the problem. *Surprise* is only one of many verbs which exhibit the so-called *property-factoring* alternation. This alternation occurs on a large class consisting of over one hundred verbs, among them

(20) amuse, anger, annoy, disappoint, embarrass, frighten, impress, please, scare, stun, ...

These verbs also share a certain semantic property: they all denote *emotional reactions*. For this reason we identify a class of *emotional-reaction* verbs and say that the property of the verb *surprise* responsible for the alternation shown in (10) and (12) holds for all verbs that comprise the *emotional-reaction* class.[3]

Once we have tied the ability to participate in the property-factoring alternation to a particular class of

[2] As shown in (11), START translates a *surprise* sentence into two T-expressions, but to simplify the exposition we do not show here the second T-expression, **<object2 related-to subject>**, describing the relation between the property (*object2*) and its possessor (*subject*).

[3] These verbs have been the subject of extensive study in the linguistic literature because of this and other characteristic properties that set this class apart. (Postal [1971], Van Oosten [1980], Pesetsky [1987], Belletti and Rizzi [1988], Grimshaw [1990], Levin [1993] and many others).

verbs, we no longer need to indicate this property in the lexical entry of each verb in the class or write verb-specific S-rules, such as the *Surprise* S-rule. Rather, we can associate the alternation with the emotional-reaction class and then simply indicate in the lexical entry of a verb whether it belongs to this class. That is, we augment a verb's lexical entry with an indication of its semantic class membership. For instance, we would register in the entry for *surprise* that it is a member of the emotional-reaction class. Now instead of writing a number of verb-specific S-rules, we can write a single general S-rule which triggers on any verb from the emotional-reaction class:

(21) *Property-factoring* S-rule
**If** <<subject verb object1> **with** object2>
**Then** <object2 verb object1>
**Provided** verb $\in$ *emotional-reaction* class

The revised S-rule contains a **Provided** clause which specifies the class of verbs to which the rule applies, ensuring that it applies to the emotional-reaction verbs. **Provided** clauses may impose restrictions on any of the S-rule variables.

When question (14) is asked, the Property-factoring S-rule (used in the backward mode) will trigger, since the T-expression <**answer surprise audience**> produced by the question matches the **Then**-part of the rule, and furthermore, the verb *surprise* belongs to the emotional-reaction class. The correct answer to question (14) is deduced when the appropriately instantiated **IF**-part of the rule is matched to T-expression (11) found in the knowledge base. Here is how START responds:

Bill's answer surprised Hillary.
*I deduced that from the following given fact:*
Bill surprised Hillary with his answer.

The **Provided** restriction of S-rule (21) not only allows the rule to apply to verbs of the appropriate semantic type, but it also prevents the rule from applying to verbs that do not display the property-factoring alternation. For instance, the verbs *surprise* and *present* can express their arguments in a similar fashion—both are found in the context [**NP V NP** *with* **NP**], but they differ in the other realizations of their arguments. Specifically *present* does not participate in the property-factoring alternation, as (22) shows, nor does *surprise* participate in the alternation that *present* participates in, as (23) shows:

(22) Hillary presented Bill with a gift.
*Hillary's gift presented Bill.

(23) Bill surprised the audience with his answer.
*Bill surprised his answer to the audience.

In the absence of the **Provided** clause, the Property-factoring S-rule could potentially misapply to verbs like *present*.

The *surprise* example shows how the addition of information about semantic class membership to verb entries allows the system to handle a particular phenomenon (or lexical property) common to all verbs in a particular class, with the help of a single S-rule. Note that the *verb class* approach allows us to extend the system to handle new properties of a class of verbs. All that is required is the addition of the appropriate S-rule, formulated so that it triggers on this class of verbs. There is no need to alter the lexical entries of the members of the class in any way as long as the lexical entry of each verb in the class indicates that it is a member of this class. Thus the verb class approach allows a more modular system design; this in turn allows the coverage of the system to be extended more easily.[4]

By expanding START's knowledge base with additional sentences and augmenting its lexicon with information about synonyms, hyponyms and additional S-rules, we allow the user to ask a larger variety of questions. Suppose that the system was given the following three statements:

Bill Clinton is the president of the United States of America.
Hillary Clinton is Bill Clinton's wife.
Bill Clinton astonished Hillary Clinton with his answer.

Now, in addition to answering questions that closely paraphrase the original statements, START will also be able to answer questions such as:

Did the answer of the president of the United States of America surprise his wife?
Was the spouse of the American president stunned by his reply?
Whose response amazed Hillary?

The examples discussed in this section show how the transparent syntax of S-rules coupled with the information about verb class membership provided by the Lexicon facilitates a more fluent and flexible dialog between the user and the language processing system.

## Natural Language Annotations

The discussion so far was centered on the analysis of single natural language sentences. We believe that given a sophisticated grammar, a large lexicon enhanced by advances in Lexical Semantics (such as class-membership information) and an inference engine (such as S-rules), it is possible to build a natural language system with satisfactory *sentence-level* performance. At the same time, however, it is becoming increasingly clear that a robust *full-text* natural language

[4] For a discussion of the system's treatment of other lexical alternations and verb classes see Katz and Levin [1988]. For a thorough classification of English verb classes and alternations see Levin [1993].

question-answering system cannot be realistically expected any time soon. Numerous problems such as intersentential reference and paraphrasing, summarization, common sense implication, and many more, will take a long time to solve to everybody's satisfaction. In the meantime, we need a mechanism that will let us bridge the gap between our ability to analyze natural language sentences and our appetite for processing huge amounts of natural language text.

The START system makes an attempt to bridge this gap by employing *natural language annotations.* (Katz and Winston [1994]). Annotations are computer-analyzable collections of natural language sentences and phrases that describe the contents of various information segments. START analyzes these annotations in the same fashion as any other sentences, but in addition to creating the required representational structures, the system also produces special pointers from these representational structures to the information segments summarized by the annotations.

Suppose, for example, that a user wishes to retrieve the following text fragment related to the discovery of Neptune:

> Neptune was discovered using mathematics. Before 1845, Uranus was widely believed to be the most distant planet. However, astronomers observed that Uranus was not always in the position predicted for it. The astronomers concluded that the gravitational attraction of a more distant planet was disturbing the orbit of Uranus.
>
> In 1845, John Adams, an English astronomer, calculated the location of this more distant planet. Urbain Leverrier, a French mathematician, independently did similar calculations. In 1846, John G. Galle and Heinrich d'Arrest of the Urania Observatory in Berlin, looked for the planet where Leverrier and Adams predicted it would be located. They saw the planet, which was later named Neptune, on September 23, 1846. Galileo, the noted Italian astronomer, may have been the first to see Neptune in 1613. However, Galileo did not recognize what he saw as a planet.

Let us assume that sentence (24) below serves as one of the annotations[5] to this text fragment:

(24) John Adams discovered Neptune using mathematics.

This means that START analyzed sentence (24) and incorporated it into the knowledge base along with a pointer to the text fragment. Now suppose the user asks one of the following questions:

(25) Who discovered Neptune?
Did Adams discover Neptune?
How was Neptune discovered?
Was Neptune discovered using mathematics?
Tell me about Neptune's discovery.

START begins the process of answering a question from (25) in the regular fashion described in previous sections. Namely, after undoing the effects of the *question* and *passive* transformations applied, START creates a T-expression such as (26) for querying the knowledge base:

(26) <*who* **discover Neptune**>

In the next step, T-expression (26) is matched against the knowledge base. It is important to emphasize that the full power of sentence-level natural language processing is brought to bear on the matching process. START's matcher works both on the *word*-level (using, if appropriate, additional lexical information about synonyms, hyponyms, IS-A trees, *etc.*) and on the *structure*-level (utilizing necessary S-rules, information on verb-class membership, nominalization *etc.*), although in the case of a very simple interaction such as (24) and (25) most of this machinery is not utilized.

Since the representational structure returned by the matcher contains a special pointer to the annotated text fragment, START's familiar sentence-level question-answering strategy is modified. Instead of passing the representational structure to the language generation system and asking it to produce an English sentence such as (24), START simply follows the pointer and presents the text fragment (HTML-marked, as appropriate) to the user.

This last presentation step can be thought of as a general procedure to be executed after a successful matching process. As a result, the natural language annotation technique easily generalizes to the indexing and retrieval of all types of information, whether or not it is based on text. Using START, one can access text, images, sound, video, web pages, and more. (Katz and Winston [1995]).

## START on the World Wide Web

In December 1993, START became the first natural language system available for question answering on the World Wide Web. The first release of the START knowledge base[6] contained information about faculty members of the MIT Artificial Intelligence Laboratory and their research. Since then START has been involved in dialogs with users all over the world, answering hundreds of thousands of questions. In response to these questions and in response to our sponsors' priorities, we expanded the original knowledge base and added new knowledge bases.

Currently, the users of various START servers can ask natural language questions about geography and climate of certain countries, weather forecasts for major cities, distances between cities, maps of numerous

[5] In the current version of the START system, most annotations are entered manually, although we are experimenting with several approaches that will make this process more automatic.

[6] It was brought to the Web with the help of the Common Lisp Hypermedia Server (Mallery [1994]).

countries and their capitals. Separately, we created a knowledge base with topical information on nuclear technology and nuclear proliferation. Another knowledge base, the Bosnia Information Server, provides access to multimedia information on the U.S. mission in Bosnia-Herzegovina. It answers questions about the geography and climate of the Bosnian region, about recent military events in the region, and about the history of Bosnian conflict.

As we added more and more information to START's knowledge base, we discovered the advantages of "virtual collaboration." We realized that the existence of the Web with its huge resources allows us to put to use the fruits of labor of a large group of people without explicitly collaborating with them. Whenever we find a new Web site with an interesting database, we identify its directory structure and argument conventions. Then we create an S-rule (or an annotation schema) which, when triggered by an English question, invokes the appropriate URL (manufactured using arguments obtained from the question) and finally presents the information to the user. At any given time, in response to a question, START can dispatch a user to a weather Web page, a map collection, a CIA database, a personal homepage, a popular search engine. It is from this "virtual collaboration" that START receives its additional power.

## Annotating the World Wide Web

The World Wide Web is a vast collection of information in digitized form, including text, relational databases, pictures, audio, video, and multi-media information. The good news in this development is that this information has been growing exponentially; the bad news is that we can make little use of it. Several problems stand in our way:

- The Web is an unstructured collection of information spanning the entire range of human experience and expression. No representation or reasoning technology we now have is capable of dealing with it.
- We can't find what we need: size and the almost complete randomness of organization make it difficult.
- The speed of growth would seem to render pointless almost any effort to keep up cataloging efforts.

So what can we do to make better use of all this knowledge? Any good researcher faced with an imposingly large and unstructured collection of information would solve the problem by simply finding someone who knows where to look. Asking a good reference librarian in the Library of Congress would be much more useful than going to Alta Vista. Notice however that the reference librarian doesn't need to *understand all the details* of the material she locates for us, only to know that it *contains relevant information.*

Hence we propose to create a smart reference librarian for the World Wide Web. Instead of attempting to capture and analyze each Web resource in detail, we will focus on more general *knowledge about that knowledge*, such as when it is relevant, to whom, and for what. We propose to attach such descriptive information to everything available on the Web. Size and speed of growth would seem to render this task impossible. But the key is to get everyone involved. To make the task of creating annotations *less* work than it's worth, we make it possible to create those annotations using a knowledge representation language that everyone knows: natural language.

By allowing thousands of people to build up knowledge about knowledge, we will create a knowledge base of an interesting form. The Web will continue to be built out of "opaque" information segments: text, maps, charts, audio, video, *etc.*; but attached to each of these will be natural language annotations that facilitate retrieval. By giving humans access to relevant information that humans can further interpret and understand, we will transform the Web into an intelligent, high performance knowledge base.

## Sample Questions Answered by START Servers

The following sample questions taken from START's log files help illustrate the range of questions asked by the WWW users.

- When was the laboratory founded?
- How can I find Marvin Minsky?
- Who at MIT performs research in arm manipulation?
- How many people do research at the AI Laboratory?
- How is the laboratory funded?
- What type of questions can you answer?
- What is the capital of Serbia?
- Where is Mostar?
- How far is it from Pale to Sarajevo?
- Is it raining in Belgrade today?
- Can I see a map of Bosnia?
- Are there American casualties in the Bosnian war?
- What did Hillary Clinton say to the US troops?
- How many mass grave sites are there in Bosnia?
- How much does the US mission in Bosnia cost?
- How many people in Serbia are Muslim?
- Do you know anything about Serbia's legislature?
- How did the Dayton agreement divide Bosnia?
- How long will the US mission in Bosnia last?

## Acknowledgements

I am grateful to Randall Davis, Wesley Hildebrandt, Thomas Marill, Michael de la Maza and Deniz Yuret for their assistance in writing this paper.

## References


Allen, J.F. and L.K. Schubert, 1991, "The TRAINS project," Technical Report 382, Department of Computer Science, University of Rochester, Rochester, N.Y.

Belletti, A. and L. Rizzi, 1988, "Psych-Verbs and Th-Theory," *Natural Language and Linguistic Theory*, 6, 291-352.

Doyle, R.J., 1984, "Hypothesizing and Refining Causal Models," M.I.T. Artificial Intelligence Laboratory Memo No. 811, Cambridge, MA.

Grimshaw, J., 1990, *Argument Structure*, MIT Press, Cambridge, MA.

Katz, B., 1980, "A Three-step Procedure for Language Generation," MIT Artificial Intelligence Laboratory Memo No. 599, Cambridge, MA.

Katz, B., 1990, "Using English for Indexing and Retrieving," In P. H. Winston and S. A. Shellard, editors, *Artificial Intelligence at MIT: Expanding Frontiers*, volume 1, MIT Press, Cambridge, MA.

Katz, B. 1994, "Using Natural Language Annotations in the Voyager Information System," *International Aerospace Congress IAC'94*, Moscow, Russia.

Katz, B. and R. Brooks, 1987, "Understanding Natural Language for Spacecraft Sequencing," *Journal of the British Interplanetary Society*, Vol. 40, No. 10, London, England.

Katz, B. and B. Levin, 1988, "Exploiting Lexical Regularities in Designing Natural Language Systems," *12th International Conference on Computational Linguistics, COLING '88*, Budapest, Hungary.

B. Katz and P.H. Winston, 1983, "A Two-way Natural Language Interface," in *Integrated Interactive Computing Systems*, edited by P. Degano and E. Sandewall, North-Holland, Amsterdam.

Katz, B. and P.H. Winston, 1994, "Method and Apparatus for Generating and Utilizing Annotations to Facilitate Computer Text Retrieval," U.S. Patent 5,309,359.

Katz, B. and P.H. Winston, 1995, "Method and Apparatus for Generating and Utilizing Annotations to Facilitate Computer Retrieval of Database Material," U.S. Patent 5,404,295.

Keshi, I. and B. Katz, 1991, "Speech-Act Based Message Conversation System," *International Workshop on Computer Supported Cooperative Work*, Berlin, Germany.

Levin, B., 1993, *English Verb Classes and Alternations: A Preliminary Investigation,* University of Chicago Press, Chicago, IL.

Mallery, J.C., 1994, "A Common Lisp Hypermedia Server," *First International Conference on the World-Wide Web,* Geneva.

Pesetsky, D., 1987, "Binding Problems with Experiencer Verbs," *Linguistic Inquiry* 18, 126-140.

Postal, P., 1971, *Cross-Over Phenomena*, Holt, Rinehart, and Winston, N.Y.

Schapiro, S.C. and W.J. Rapaport, 1987, "Sneps Considered as a Fully Intensional Propositional Semantic Network," in *The Knowledge Frontier*, edited by N. Cercone and G. McCalla, Springer-Verlag, N.Y.

Van Oosten, J., 1980, "Subjects, Topics and Agents: Evidence from Property-factoring," *Proceedings of the Berkeley Linguistics Society* 6, Berkeley, CA.

Warren, D.H.D. and F.C.N. Pereira, 1982, "An Efficient Easily Adaptable System for Interpreting Natural Language Queries," *American Journal of Computational Linguistics*, vol. 8, num. 3-4.

Winston, P.H., 1992, *Artificial Intelligence*, Addison-Wesley, Reading, MA.

Winston, P.H., 1982, "Learning New Principles from Precedents and Exercises," *Artificial Intelligence*, vol. 19, no. 3.

Winston, P.H., T.O. Binford, B. Katz, and M.R. Lowry, 1983, "Learning Physical Descriptions from Functional Definitions, Examples, and Precedents," *National Conference on Artificial Intelligence*, Washington, D.C.